AFRL-IF-WP-TR-2005-1573

AEROSPACE SENSOR COMPONENT AND SUBSYSTEM INVESTIGATION AND INNOVATION-2 COMPONENT EXPLORATION AND DEVELOPMENT (ASCSII-2 CED)

Delivery Order 0005: Reconfigurable Processing With Bio-Inspired Embedded Systems On A Chip

Dr. Hoda S. Abdel-Aty-Zohdy

Systran Federal Corporation
4027 Colonel Glenn Highway, Suite 210
Dayton, OH 45431-1672

OCTOBER 2005

Final Report for 02 July 2003 – 02 January 2005

Approved for public release; distribution is unlimited.

INFORMATION DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7334

NOTICE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Wright Site (AFRL/WS) Public Affairs Office (PAO) and is releasable to the National Technical Information Service (NTIS). It will be available to the general public, including foreign nationals.

PAO Case Number: AFRL/WS-05-2114, 12 Sep 2005

THIS TECHNICAL REPORT IS APPROVED FOR PUBLICATION.

_____/SS/_____ _____/SS/_____
[signature of project monitor/in-house author] [signature of supervisor]

_____/SS/_____
[signature of three letter chief]

This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
October 2005	**Final**	**02 Jul 2003 – 02 Jan 2005**

4. TITLE AND SUBTITLE

Aerospace Sensor Component And Subsystem Investigation And Innovation-2 Component Exploration And Development (ASCII-2 CED)

Delivery Order 0005: Reconfigurable Processing With Bio-Inspired Embedded Systems On A Chip

5a. CONTRACT NUMBER
F33615-00-D-1726-0005

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62204F

6. AUTHOR(S)
Dr. Hoda S. Abdel-Aty-Zohdy

5d. PROJECT NUMBER
2002

5e.TASK NUMBER
06

5f. WORK UNIT NUMBER
08

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Systran Federal Corporation
4027 Colonel Glenn Highway
Suite 210
Dayton, OH 45431-1672

Microelectronics System Design Lab
Department of Electrical and Systems Engineering
Oakland University
Rochester, MI 48309-4401

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
INFORMATION DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AFB, OH 45433-7334

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/IFTA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-IF-WP-TR-2005-1573

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
This document contains color.

14. ABSTRACT

The idea of exploring the multidimensional characteristics biocomputing deals with involves a multilevel *protein* logic interaction with hybrid analog and digital systems (bio/digital/analog systems). In the *protein* biocomputing world, all models are nonlinear exponential in nature. These nonlinear systems are linearized by boundary limited frequency, time response and dimensional ranges. The protein model and its *correct* development is probably the most difficult aspect for biocomputing issues. These future computing devices will be chemical sensors, fluidic computer systems, optical switches, nanoactuators, self-repairing computing systems, nanorobots and reusable analog components. Noise reduction techniques for these systems involve spiking neural network techniques for data extraction. Reusable Analog Integrated Circuit Components are needed for hardware signal processing and for System-on-A-Chip (SoC) Integration. These include: Microwave Operational Amplifier(s); Switched Capacitor Filters; and Voltage References. During this research period, the analog design efforts have been focused on re-configurable microwave Op-Amp. A prototype 0.18 CMOS SOI design structure fort RF high-gain op-amp was submitted for prototype implementation.

15. SUBJECT TERMS:
Biocomputing, protein model, analog, integrated circuits, system-on-a-chip and chemical sensors

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**Robert L. Ewing**
Unclassified	**Unclassified**	**Unclassified**	**SAR**	**55**	19b. TELEPHONE NUMBER *(include area code)* **937- 255-6653 X3592**

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

Table of Contents

Executive Summary

This is the final report of the research project entitled "Reconfigurable Processing with Bio-Inspired Embedded Systems On a Chip (SoC) Architectures for Aerospace". During this period, the project has supported three graduate students (Jacob Allen on the Spiking Neural Network, Paul Wierzbicki and Lem Liou on Protein Modeling) and the Principle Investigator, Dr. Hoda S. Abdel-Aty-Zohdy, to study the three technical problems related to bio-inspired SoC.

The idea of exploring the multidimensional characteristics biocomputing deals with involves a multilevel *protein* logic interaction with hybrid analog and digital systems (bio/digital/analog systems). In the *protein* biocomputing world, all models are nonlinear exponential in nature. These nonlinear systems are linearized by boundary limited frequency, time response and dimensional ranges. The protein model and its *correct* development is probably the most difficult aspect for biocomputing issues. These future computing devices will be chemical sensors, fluidic computer systems, optical switches, nanoactuators, self-repairing computing systems, nanorobots and reusable analog components. Noise reduction techniques for these systems involve spiking neural network techniques for data extraction. Reusable Analog Integrated Circuit Components are needed for hardware signal processing and for System-on-A-Chip (SoC) Integration. These include: Microwave Operational Amplifier(s); Switched Capacitor Filters; and Voltage References. During this research period, our analog design efforts have been focused on re-configurable microwave Op-Amp. A prototype 0.18 CMOS SOI design structure fort RF high-gain op-amp was submitted for prototype implementation through the MIT Lincoln lab, FDSOI process.

A. Technical Results

Our research has been on three topics; One is a recursive spiking attractor neural network, the second is in protein memory modeling and the third is reconfigurable CMOS analog. The following work is based on Jacob Allen's thesis work at Oakland University and work within the Air Force Research Laboratory, Information Directorate.

A.1 Recursive Spiking Attractor Neural Network

Scientists have long sought to model the brain and unlock its secrets. Understanding brain dynamics, how-humans-think, is a holy grail of science that is still a mystery after sixty years of diverse research. The body of neural related knowledge spans scientific disciplines from psychology to finance. The motivations of neural network research can be classified broadly into two categories. In one category, researchers are attempting to understand and explain biological neural systems. In the second category, researchers are attempting to solve specific problems using techniques inspired by biological systems. This research fits squarely in the second category, because it focuses specifically on problems related to artificial olfaction. However, the thesis was strongly influenced by research that fits in the first category. Physicists attempting to model biological brain activity have been aided by improvements in brain imaging technology. Complex models of spiking neural networks developed by physicists help to explain certain brain dynamics and may eventually provide a mechanism for understanding and duplicating animal intelligence at a macroscopic level. For example, in one instance, learning phenomena observed in monkey cortexes was roughly duplicated using biologically plausible networks[1].

A.1.1 Neurons in Biology

Biology is the main source of inspiration for research in neural networks. Millions of years old, the parallel structure of the animal brain can still manage tasks that are far beyond the capabilities of the world's fastest super computers. Neurons are the main processing units of the brain. A biological nerve cell, or neuron, is composed of three main components. The axon receives signals from other connected nerve cells, the cell body contains the cell, and the dendrites send signals from the neuron to other connected neurons. The synapse is the junction where the dendrite of a transmitting neuron connects with the axon of a receiving neuron. It is common to refer to the neuron that transmits a signal as *pre-synaptic*, and refer to the neuron that receives a signal as *post-synaptic*. In general, biological synapses operate as chemical junctions. When a presynaptic neuron fires, it releases positively charged ions from its dendrites.

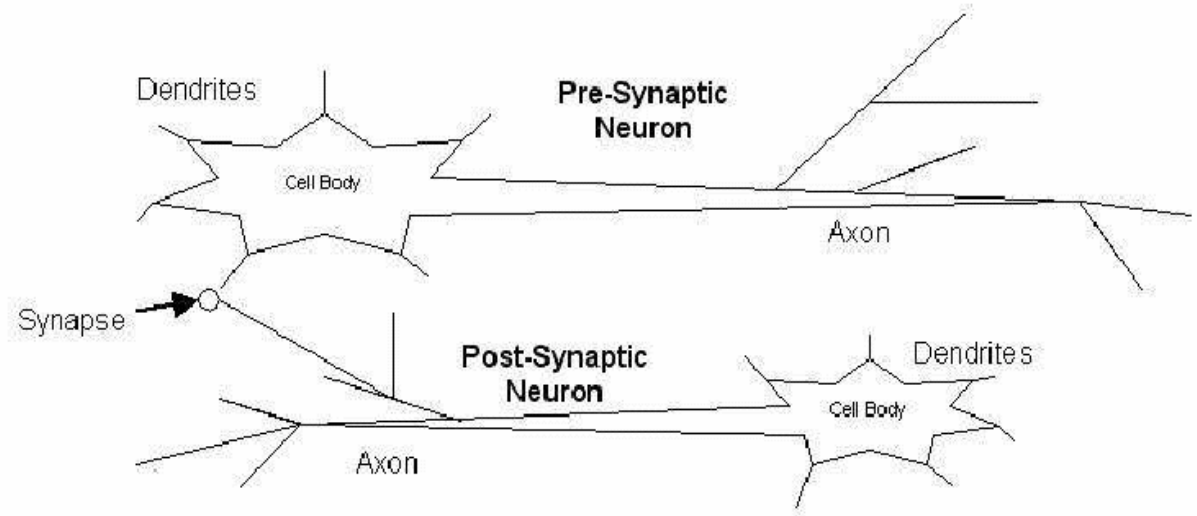


Figure A.1.1: Basic Neuron Structure

The ions diffuse to the axon of the post-synaptic neuron and collect over a period of time. Once the post-synaptic neuron collects enough ions, a voltage difference accumulates between the dendrites and axons, and the neuron becomes a positively charged capacitor. Eventually, the charge builds up to a threshold voltage, where-upon the neuron emits an electrical spike that moves from the axon to the cell dendrites. This spike is released through the dendrites as charged ions, and the cycle continues. The spiking behavior of neurons is observable as brain wave patterns that can be studied in biological brains. Techniques such as Magnetic Resonance Imaging (MRI) allow researchers to probe and record the spiking activity in the brain. In one experiment, Miyashita studied spiking patterns in the pre-frontal cortex of the monkey's brain, an area long associated with sensory vision processing[1]. Pairs of fractal images were shown to a monkey, one after the other, with a delay between. When two images matched, the monkey could press a lever to receive a reward for remembering the image. MRI technology monitored the spiking behavior of individual nerve cells in the monkey's cortex while it performed this task. A pattern emerged in the spike activity. When the monkey was not stimulated by a recognition task, the neurons fired at a low background spike rate. However, when the monkey recognized certain patterns, isolated neurons would spike at a much faster rate. Finally, these neurons maintained a higher spiking rate for a period of time after the presentation.

A.1.2 Modeling Brain Activity

Models of brain activity start with models of neuron activity. Basic neuron models developed by biologists and physicist are presented in this section.

A.1.2.1 The LIF Neuron

The linear integrate and fire (LIF) neuron is the primary element of most biological neural network models. At a macroscopic level, the LIF neuron captures the essence of well understood biological neuron dynamics. In this model, a neuron has an internal variable, $V(t)$, that tracks the voltage potential between the neuron's axon and dendrites. $V(t)$ is increased when spikes from external neurons are received at the axons. The amount of

change in $V(t)$ for each spike received is defined by a synaptic weight, J. Once $V(t)$ reaches a certain threshold, θ_v, the neuron depolarizes by emitting a spike, and simultaneously resets the internal voltage so that $V(t) = 0$. For a short period of time after depolarization, defined by τ_{arp}, $V(t)$ remains at the 0 voltage level. When the neuron is not receiving spikes, $V(t)$ decreases by a linear decay factor, β_v. However, $V(t)$ cannot be negative, and $V(t)$ cannot decay below 0 volts.

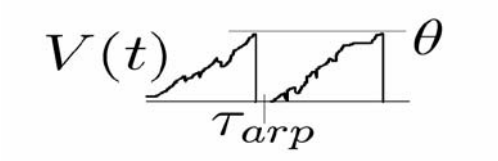


Figure A.1.2: Example of Neuron Depolarization

$V(t)$ gradually increases, until it reaches μ. After depolarization, there is a delay,

$$\frac{dV(t)}{dt} = \begin{cases} -B_v + I(t) \; when \; 0 < V(t) < \theta \\ 0 \; when \; V(t) < 0 \; or \; V(t) > \theta \end{cases} \qquad Eqn\ 1.1$$

$I(t)$ is the influx current which is generated from input spikes, and is discussed in a following section.

A.1.2.2 Synapses

Synaptic connections may be either excitory or inhibitory. Spikes received at the neuron via excitory synapses increase $V(t)$, while spikes received via inhibitory synapses decrease $V(t)$. Because $V(t)$ cannot be negative, inhibitory spikes cannot drive $V(t)$ below 0 volts. There is modest debate between biologists concerning the weights of synaptic connections. In one camp the theory is that synaptic connections are either on or off. Experimentation indicates that synaptic connections between neurons are all or nothing[5]. Spikes received via a depressed synapse contribute J_d to $V(t)$, while spikes received via a potentiated synapse contribute J_p.

In the other camp, some biologists argue that there is no evidence to disprove the theory that synapse weights vary on a sliding scale of many possible weights. This deep synapse theory allows the weights of synapses to be finely adjusted during the learning process. Most of the conventional neural network models described in [4] use deep synapses as the foundation of their mathematical models. When modeling networks, the choice of binary synapses or deep synapses is application specific. Amit shows that biologically plausible networks using binary synapses can learn patterns as well as those using deep synapses[6]. Additionally, binary synapses help to reduce noise and add to stability in a network. For our purposes, binary synapses make the model easier to understand. Further, binary synapses seem easier to implement in VLSI than deep synapses.

A.1.3 Recurrent Networks of LIF Neurons

Begin with a very large network of neurons that are randomly and recurrently interconnected. A recurrent spiking neural network is typically composed of three neuron populations. First, there is a foreground population of N neurons that are randomly interconnected with one another. Second, there is an inhibitory neuron population with N_{inh} neurons that is connected to the foreground population through inhibitory synapses. Finally, an external population of N_{ext} background neurons always fires with a constant mean rate, ν, and is randomly connected to the foreground population.

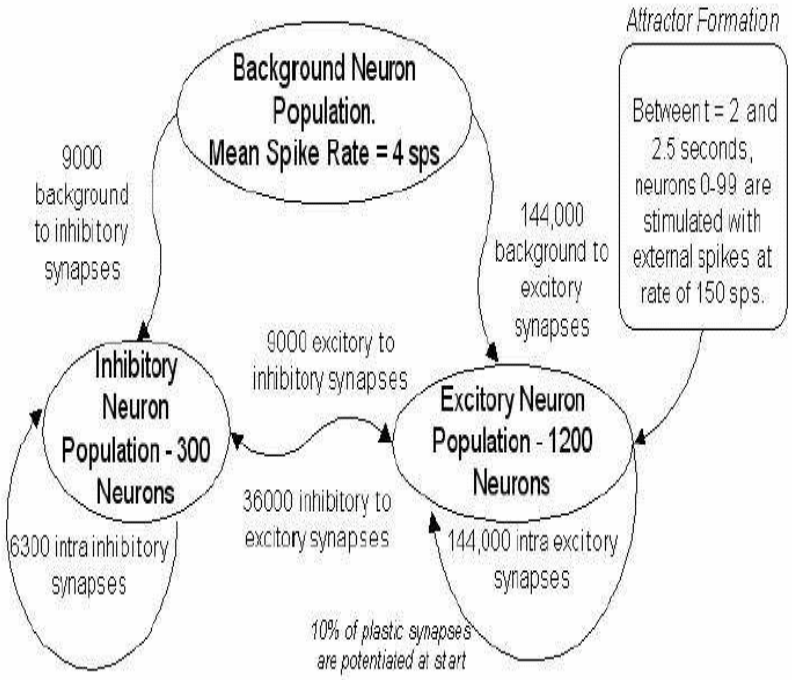


Figure A.1.3: Typical Recurrent Neural Network
Populations of Inhibitory, Excitory, and Foreground neurons are randomly connected by binary synapses.

A diagram of a typical network configuration simulated for this research is show in Figure A.1.3. Meanfield theory describes the dynamics of randomly interconnected networks with sparse connections[2]. The theory essentially calculates the probability that a neuron in the network will spike at any given moment, and extends this probability to describe the dynamics of the entire network. The result is a Gaussian probability density function that describes the influx current to a neuron, $I(t)$ in terms of a mean influx υ_i and standard deviation β_i. Define c as the count of synapses connected to a neuron. When $I(t)$ is injected from an external population, the parameters if the Gaussian density function are given in [7].

$$\upsilon_i = \nu \bullet c \bullet N \bullet J \qquad \textit{Eqn 1.2}$$

$$\sigma_i^2 = \nu \bullet c \bullet N \bullet J^2 \qquad \textit{Eqn 1.3}$$

A solution for $v(t)$, defined as the mean spike rate, is given in terms of μ_i and σ_i [2].

$$v(t) = [\tau_{arp} + \frac{\sigma^2}{2\mu^2}(\frac{2\mu_i\theta}{\sigma_i^2} - 1 + e^{\frac{-2\mu_i\theta}{\sigma_i^2}})]^{-1} \qquad \textit{Eqn 1.4}$$

Neurons in the foreground are also affected by feedback within their own population. Therefore, $v(t)$, is a function of itself, and contributes to μ_i and σ_i.

$$\mu(t) = a_u v(t) + b_u(t) \qquad \textit{Eqn 1.5}$$

$$\sigma^2(t) = a_\sigma v(t) + b_\sigma(t) \qquad \textit{Eqn 1.6}$$

Here, a_μ and a_σ represent the random contribution of $v(t)$, while b_μ and b_σ represent the external contribution to μ_i and σ_i. Therefore, a recurrent network of neurons will fire at a fixed rate $v(t)$, when Equation 1.4 has a solution[2]. Nonlinearities in Equation 1.4 allow for more than one solution, so the network can be stable at multiple spike rates. In the sections that follow, we will discuss how spike rates are used in memory storage.

A.1.4 Network storage

Binary patterns, or prototypes, are stored and recalled by the network through spike rates. To understand the network recall mechanism, consider a fourth component, stimulus. A neuron in the foreground population is stimulated by linearly increasing $V(t)$ with a constant value, over time, until the neuron spikes at a rate of v_{stim} spikes per second. Using the method described in [2], a prototype pattern of n binary inputs is introduced to the network by assigning neurons to each input and stimulating neurons that correspond to `1' in the binary pattern. Hereafter, the neurons that are stimulated by a binary prototype will be referred to as prototype neurons. When a pattern is recalled by the network, prototype neurons continue to spike at a stable rate of v_{on} spikes per second, even when the stimulus is removed. However, neurons that do not belong to the recognized pattern revert back to the mean spike rate v_{off} once the stimulus is removed. This higher-than-background spike rate, referred to as selective delay activity, persists until another pattern is presented to the network. Mechanically, a pattern is learned in the network when most of the random synaptic connections between prototype neurons are potentiated, and most of the connections between prototype neurons and non-prototype neurons are depressed. Essentially, prototype neurons have non-random connections that allow them to respond to a stimulus differently than other neurons. These non-random connections between prototype neurons create a localized basin of attraction within the network.

A.1.5 Learning

Learning is the process of arranging synaptic connections to form basins of attractions that will correspond to learned prototypes. Perhaps learning is the least understood component of biological neural networks. Hebb proposed long ago that learning is a function of the spike rates in the pre and post synaptic neurons. The general theory is that when two neurons are active at the same time, the synaptic strength between them increases. When only one of the two neurons is active, the connection strength of the synapse decreases. In a system with binary synapses, Hebbian learning can only cause a synapse to become potentiated or depressed. Fusi has proposed a new Hebbian learning mechanism compatible with bistable recurrent neural networks[2]. In this network, synapses are plastic, which

signifies their ability to change state from a potentiated state to a depressed state, and vice versa. In this learning model, each synapse has an internal variable, $X(t)$, that tracks the spiking activity of the pre and post-synaptic neurons as shown in Figure A.1.5, Figure A.1.6, and Figure A.1.7. $X(t)$ varies between 0 and X_{max}. When $X(t) > \theta_x$, the plastic synapse is potentiated. When $X(t) < \theta_x$, the plastic synapse is depressed. The internal variable $X(t)$ is modified by Hebbian learning dynamics. In Fusi's plastic synapse, Hebbian learning is driven by the pre-synaptic spike train. Each time the pre-synaptic neuron transmits a spike across the synapse, the internal variable $X(t)$ is updated according to the state of $V(t)$ in the post-synaptic neuron. When $V(t)_{post} > \theta_x$, $X(t)$ is adjusted upward by a small amount, β. If $V(t)_{post}$ in the post-synaptic neuron is below μv when the pre-synaptic neuron transmits a spike, then the synaptic variable $X(t)$ is adjusted downward by a small amount, $\bar{}$. This dynamic effectively implements a stochastic Hebbian learning law. A refresh mechanism also exists to preserve the long term steady state values of $X(t)$. When the synapse is idle, a dampening/refresh factor causes $X(t)$ to drift towards either 0 or X_{max} over the long term steady state. When $X(t)$ is greater than θ_x, $X(t)$ is linearly adjusted upward with respect to time by a factor, If $X(t)$ is less than θ_x, then $X(t)$ is linearly adjusted downward by a factor, β. The refresh adjustment is limited by the barriers of 0 and X_{max}; so, in effect, the internal state variable will move towards a steady state of 0 or X_{max}, and θ_x is the dividing point. Therefore, $X(t)$ may be described in terms of a refresh component $R(t)$ and a Hebbian component $H(t)$[8].

$$\frac{dX(t)}{dt} = R(t) + H(t) \qquad \textit{Eqn 1.7}$$

$$R(t) = -\alpha \bullet \Theta(-X(t) + \theta_x) + \beta \bullet (X(t) - \theta_x) \qquad \textit{Eqn 1.8}$$

(where Θ = Heaviside function)

$$H(t) = \begin{cases} a \bullet C_{pre}(t) when V_{post}(t) > \theta_v \\ -b \bullet C_{pre}(t) when V_{post}(t) < \theta_v \end{cases} \qquad \textit{Eqn 1.9}$$

$C_{pre}(t)$ represents the number of spikes that are received from the pre-synaptic neuron at time t. $V_{post}(t)$ represents the voltage of the post-synaptic neuron at time t. Here, a is the incremental constant increase in $X(t)$ from a spike received during learning, and b is the incremental constant decrease in $X(t)$ from a spike received during learning. $X(t)$ has reflecting barriers at $X = 0$ and $X = X_{max}$ so it is held inside the range $[0; X_{max}]$.

A.1.5.1 Learning with Plastic Synapses

Learning in a network of LIF neurons connected by plastic synapses is a stochastic process, because it is based on the probabilities that a synapse will transition from long term potentiation (LTP) to long term depression(LTD), or vice versa. The transition probabilities are related to the spike rates of the pre-synaptic neuron and the post-synaptic neuron. First, it should be noted that the Hebbian learning term, $H(t)$, is dependent upon the probability that $V(t) > \theta_v$, defined as Q_a. Q_a is indirectly related to the post-synaptic neuron's spiking rate, and it may be calculated directly if the statistics of the input currents are known[2].

The probability that a synapse will transition from depressed to potentiated state is referred to as P_{LTP} (probability of long term potentiation); and the probability that a synapse will transition from a potentiated state to a depressed state is referred to as P_{LTD} (probability of long term depression). P_{LTP} and P_{LTD} may be calculated with a Takacs Process that describes the spike train and from the numerical solution to a complicated multivariate probability distribution function in partial differential equation form[2].

Thus, synapses become potentiated or depressed when neurons in the network are stimulated at a higher rate than the background spike rate. When exposed to the stimulus of

a binary prototype, these changes in synaptic states lead to non-random connections between prototype-neurons, and result in attractor formations. The learning process can be controlled by adjusting synapse parameters, which will change the transition probabilities of different sets of spike frequencies[2].

Learning is fast when the transition probabilities are high, but the depth of the memory is low because forgetting will be fast as well. Learning is slow if the transition probabilities are low, but the memory capacity is larger. It has been shown that a memory based on plastic synapses can have optimal storage capacity, and the memory will not overload once the capacity is reached[9]. Such memories act like a palimpsest, where a sliding window of the most recent memory is maintained and the older memories are slowly forgotten. Finally, once an attractor forms, the residual high spiking rate of an excited attractor, ν_{on} will act as a stimulus that increases P_{LTP} , and may lead to an associative memory between patterns associated in time. This behavior may begin to explain the Myashita experiments where monkeys showed a residual spiking rate in selected neurons after recognizing a series of two images[2].

A.1.5.2 Learning Statistics

Before moving on to my own simulations of Fusi's network, a final note about learning capacity and speed is in order. The sparseness of a binary stimulus, f, is the ratio of the number of neurons stimulated, n, to the number of neurons in the foreground network, N.

$$f = \frac{n}{N} \qquad \textit{Eqn 1.10}$$

The network storage capacity is related to f, P_{LTP} , P_{LTD}, and the timing of binary pattern presentation. When the inputs to the network are independent of N, the maximum number of patterns that can be stored is $log(N)$[2,6]. However, when the inputs are sparse, meaning that f is small, the network capacity improves. The optimal storage capacity occurs when $f \sim \frac{\log N}{N}$ [2].

A.1.6 VLSI

Spiking networks based on LIF neurons and plastic synapses are well suited for analog VLSI. They are basically immune to noise introduced in the circuit by the manufacturing process and the external environment. Randomness introduced by a factor such as temperature enhances the stochastic properties of the network. For example, the plastic synapse can be implemented as an analog circuit in VLSI with very little surface area[3]. describes how the circuit in Figure A.1.9 was implemented in Standard Analog CMOS 1:2u technology using about 90u x 70u surface area. In that circuit, the capacitor voltage acts as the synapse state variable, $X(t)$, and currents are injected or released from the capacitor when the Hebbian circuit is stimulated by a pre-synaptic spike. The refresh circuit maintains the steady state values in the synapse, while the dendrite circuit passes current to an LIF neuron when a spike is received[8]. This VLSI implementation is intriguing, because it implements on-chip learning using an algorithm that is localized to each synapse. As in biological networks, learning does not require complicated feedback circuitry.

The network described therein operates on binary patterns similar to those observed in the glomeruli of olfaction bulbs. The attractor dynamics model behavior observed in the olfactory cortex. The learning dynamics are localized to each synapse, and there is no need

for unwieldy feedback circuitry. Finally, this system has already been modeled in VLSI by other investigators.

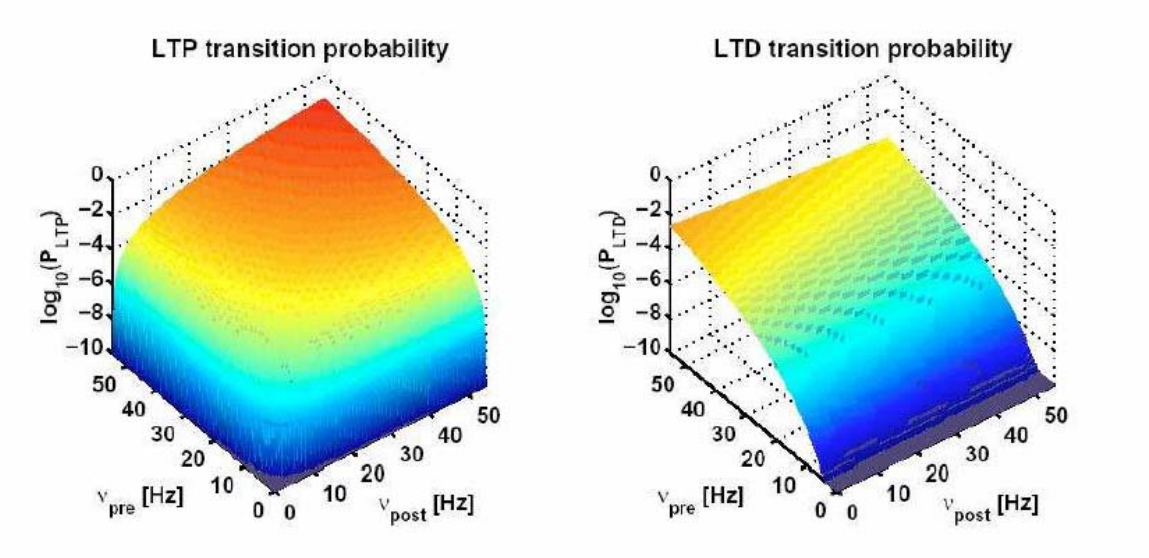


Figure A.1.4: Example of Synaptic Transition Probabilities

Biologically plausible synaptic transition model described in [8]. Long term potentiation probability varies with pre-synaptic and post-synaptic spike rates, while Long term depression probability varies with pre-synaptic and post-synaptic spike rates. Note: From Spike-Driven Synaptic Plasticity: Theory, Simulation, VLSI Implementation," by Fusi et. al., 1998, Neural Computation, 12. Copyright 2000 by the Massachusetts Institute of Technology. Reprinted with permission.

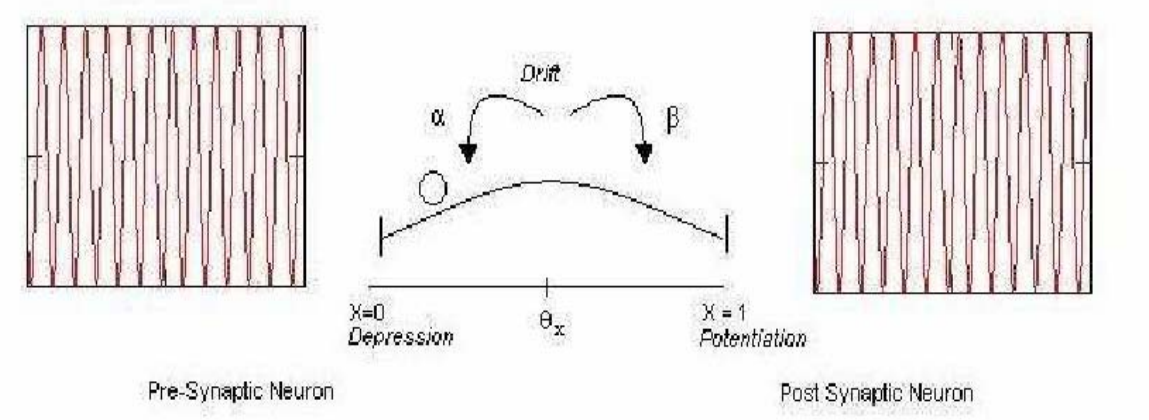


Figure A.1.5: Dynamics of Synapse State Variable $X(t)$ - Low Activity
Both the pre and post synaptic neurons are firing at the mean background rate. X(t) just drifts to long term state of 0 or X_{max} depending upon which side of the θ_x barrier it is on.

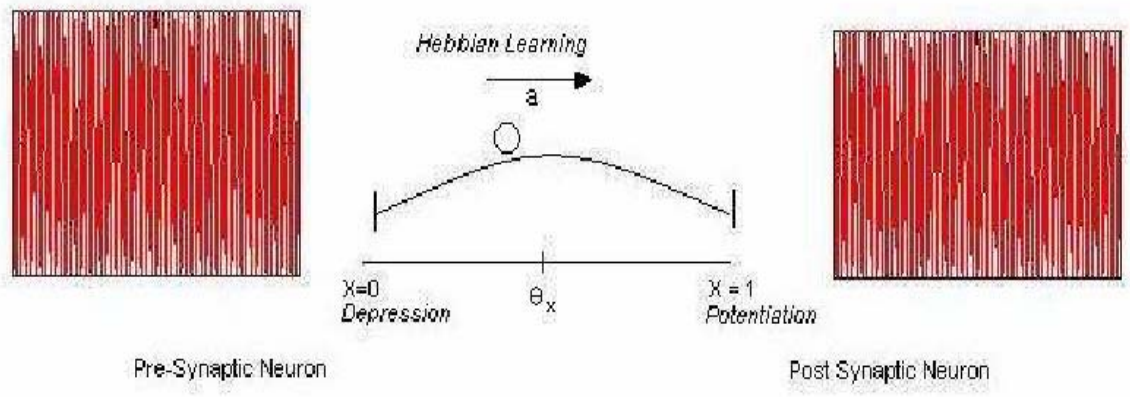


Figure A.1.6: Dynamics of Synapse State Variable $X(t)$ - Potentiation
Both the pre and post-synaptic neurons are firing at a high rate. X(t) increases by Hebbian learning term a. The probability that the synapse will transition from long term depression to long term potentiation is high.

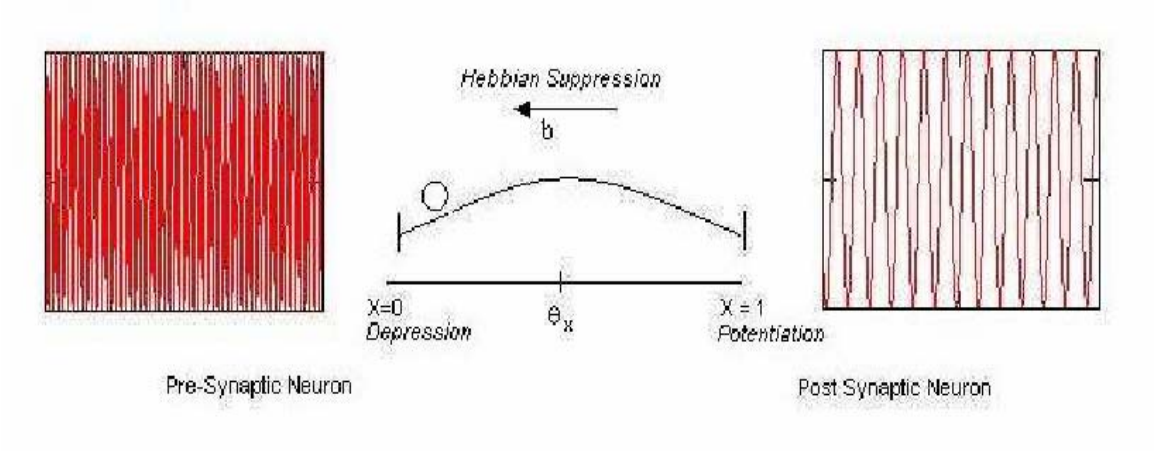


Figure A.1.7: Dynamics of Synapse State Variable $X(t)$ - Depression
The pre-synaptic neuron is firing at a high rate but the post synaptic neuron is firing at a low rate. X(t) is adjusted downward by Hebbian learning term B. The probability that the neuron will transition from long term potentiation to long term depression is high.

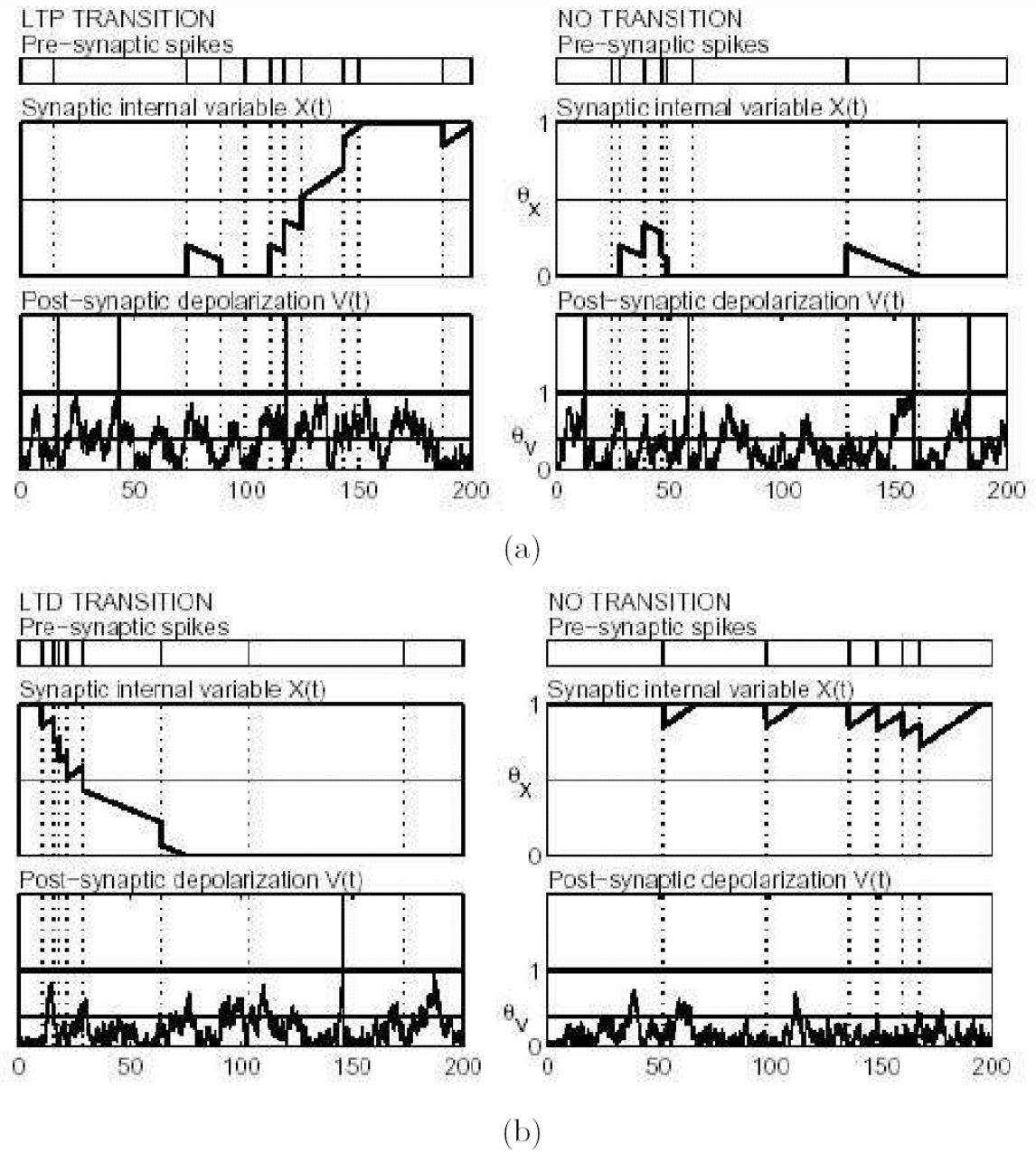


Figure A.1.8: Plastic Synapse Transition Examples

Plot (a) shows pre-synaptic spikes, V (t), and X(t). First, the synapse transitions from depressed state to potentiated state when X(t) crosses threshold, µx. Next, the synapse remains depressed, because X(t) decays to 0. (b) First, X(t) falls below µx, causing synapse to transition from potentiated state to depressed state. Next, X(t) remains at Xmax, and synapse stays potentiated. Note: From Spike-Driven Synaptic Plasticity: Theory, Simulation, VLSI Implementation," by Fusi et. al., 1998, Neural Computation, 12. Copyright 2000 by the Massachusetts Institute of Technology. Reprinted with permission.

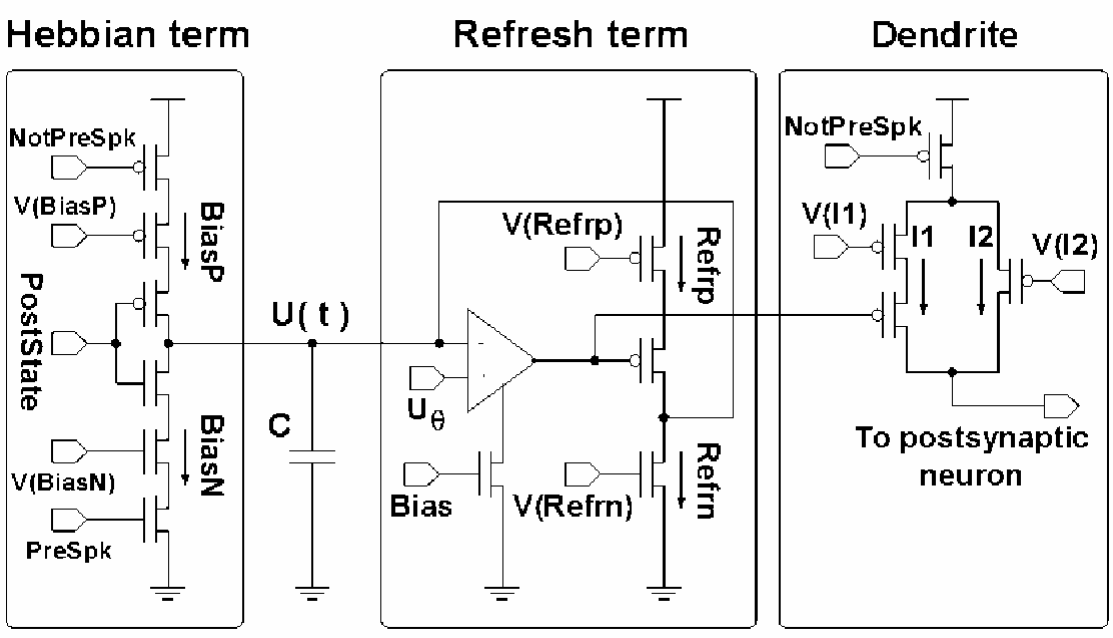


Figure A.1.9: Analog VLSI Implementation of Plastic Synapse
For details, see Fusi [8]. Note: From \Spike-Driven Synaptic Plasticity: Theory, Simulation, VLSI Implementation," by Fusi et. al., 1998, Neural Computation, 12. Copyright 2000 by the Massachusetts Institute of Technology. Reprinted with permission.

A.2. Simulation of Recurrent Spiking Networks

To better understand Fusi's network, we created two prototype simulators of recurrent spiking neural networks with plastic synapses.

A.2.1 SQL Based Recurrent Network Simulator

A simulator of LIF neurons with plastic synapses was created using *SQL Server*, which is an ideal environment for handling very large sets of data in an optimal fashion. *SQL Server* was also useful for rapid prototyping, because it is a high level language that automatically handles things like memory allocation. The simulation routine was created as a stored procedure, and tables were used to store simulation data. Data tables proved to be very useful for both debugging and for data analysis. *SQL server* was also much faster than similar experiments using *Matlab* simulations. The simulation speed was further increased by using optimization techniques suggested by[10]. Throughout this thesis, this SQL based recurrent network simulator will be referenced as the SQL Recurrent Network Simulator (SRNS).

The main goals of the SQL Recurrent Network Simulator were:

- Simulate the random spiking characteristics of an LIF network
- Simulate very basic attractor formation in the network

The simulated network consisted of 1200 LIF neurons, 300 inhibitory neurons, and a background population of neurons firing randomly at a mean rate of 8 spikes per second, as

shown in Figure A.1.3. Each neuron received 30 inputs from the inhibitory neurons, 120 inputs from other LIF neurons, and 120 inputs from outside neurons. Approximately 144,000 synapses between LIF excitory neurons were plastic, with 90% of these synapses starting in a long term depressed state. This network configuration was originally suggested as being biologically plausible by Amit [6] and is typically used throughout the literature. In the simulation, each neuron received an identification number. Neurons 0-1199 were excitory neurons, and neurons 1200-1499 made up the inhibitory population. For the purpose of analysis, neurons were collected into groups of 100. Thus, group 0 includes neurons 0-99. Group 100 included neurons 100-199, and so forth. The synaptic connections between the neurons were randomly selected so that each neuron received 30 inhibitory inputs and 240 excitory inputs.

1. The first step involved tuning the network parameters so that the mean spike rate would be approximately 4 spikes per second (sps). A rate of 3.2 sps resulted, as shown in Figure A.2.1. This met the first objective of random, steady-state spiking.
2. In step two of the experiment, a subset of the neurons was stimulated for 500*ms*. The network was allowed to run in its steady state for the first 2s. After 2 s, neurons 0-99 were excited by increasing the mean background firing rate from 4 sps to 150 sps.

As expected, the probability of synapse transition was higher for these neurons spiking at the higher rate, and a percentage of the synaptic connections local to neurons 0-99 changed from LTD to LTP. This subpopulation of connected neurons participated in the beginning of an attractor formation. Figure A.2.1 shows the mean spiking rates of the network. The spiking rate slowly increased until steady-state was reached at $t = :5s$. The foreground excitory neurons (0-1199) showed a mean spike rate of about 3.2 sps, while the inhibitory neurons (1200-1499) showed a mean spike rate of about 13 sps. At time $t = 2$ s, a stimulus was applied to neuron group 0-99, and a higher spike rate of 30 sps resulted. It is also interesting to observe that the inhibitory neuron spike rate also increased to compensate the higher excitory spike rate. The net result was that non-stimulated neurons (100-1199) showed a slightly lower spike rate of about 1 sps during the stimulation period.
The final state of the network after simulation is shown in Figure A.2.2, demonstrating that the synaptic connections within the 0-99 block were higher than the average. This indicated the beginning of an attractor formation. However, for true attractors to form in the network, one would expect LTD transitions in the synaptic connections between neuron group 0-99 and neurons 100-1199. Here, the opposite was true and an increase in LTP transitions is actually observed in Figure A.2.3. This is due to the selected network parameters for this simulation.

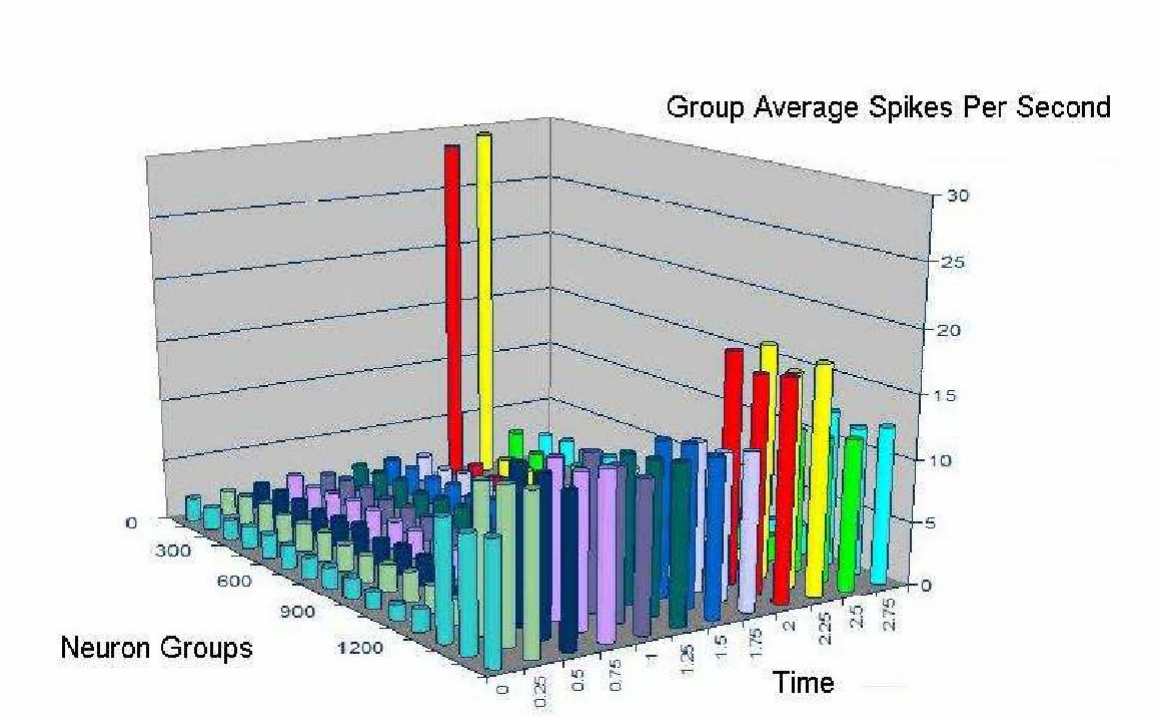


Figure A.2.1: Spike Rate by Neuron Group and Time Interval Spikes per second (vertical axis) sorted by neuron groups of 100 (left axis) and .5 second time intervals. (right axis). *Neurons 1-1199 are excitory neurons and spike with rate of about 4 sps. Neurons 1200-1499 are inhibitory neurons and spike with a mean rate of about 12 sps. Stimulus was applied to excitory neurons 1-99 at time 2 s increasing the spike rate to 29 sps.*

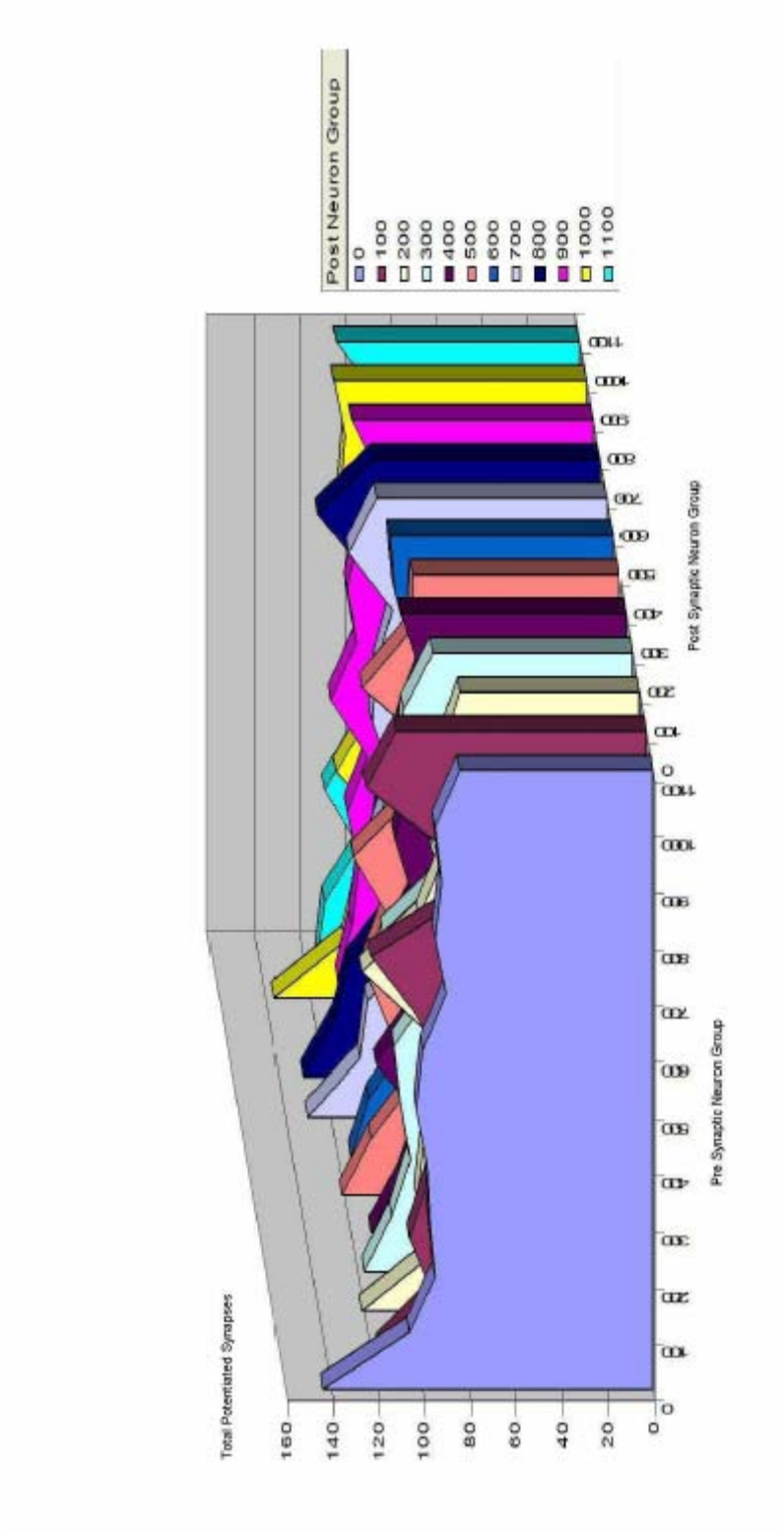


Figure A.2.2: Synapse Potentiation

Resulting potentiated synapses after stimulus. As expected, connections between stimulated neurons in group 1-99 have the most potentiated synapses after stimulus. Potentiated connections are also increased for synapses having neurons 1-99 as pre-synaptic neurons.

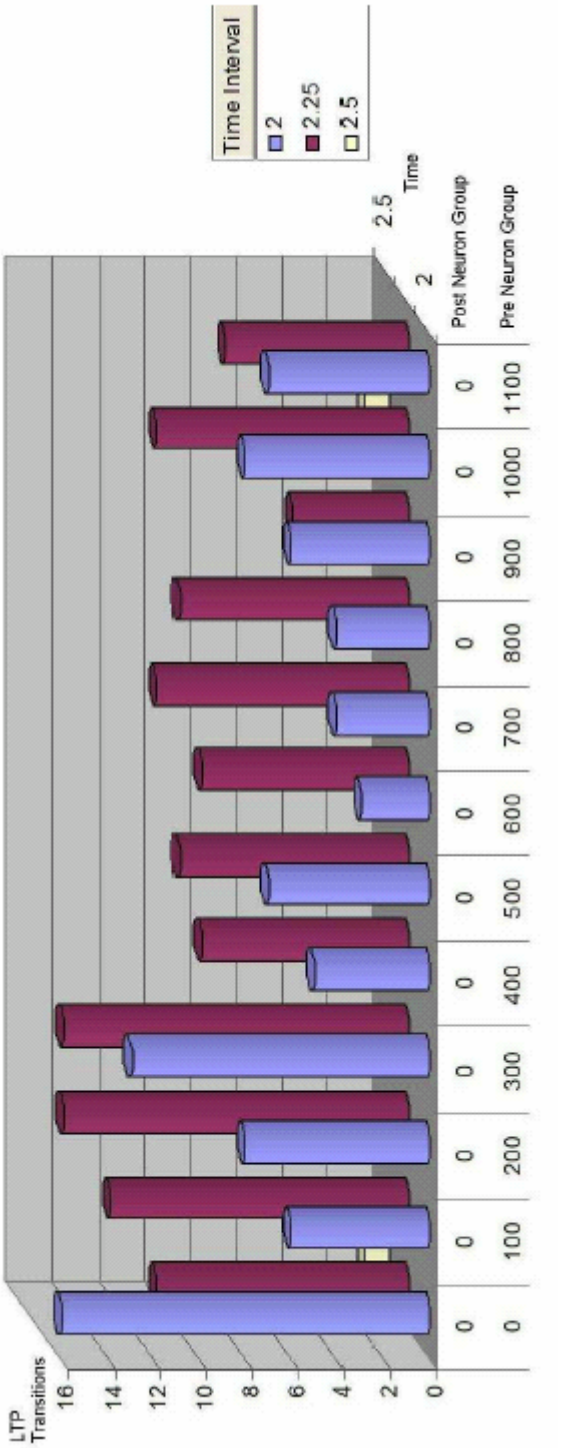


Figure A.2.3: Total Synaptic Transitions Due to Stimulus.
Synaptic transitions between stimulus time 0 seconds to time 2.5 seconds, for synapses leading from neuron group zero to the pre-synaptic groups 0 to 1100. The stimulated pre-synaptic group 0 showed a larger than average number of transitions

The SQL Recurrent Network Simulator was able to simulate large populations of neurons and demonstrate basic attractor formation. However, the simulator was slow. It took about 30 minutes to simulate one second of network activity on a 1.4 Ghz Pentium system with 256 Mb of memory. Further, there was no user interface, and the stored procedure code grew large and unwieldy.

A.2.2 Object Oriented Recurrent Network Simulator

A second, object oriented recurrent network simulator was created in using C# to overcome the weaknesses of first the SQL recurrent network simulator. A class hierarchy of the simulator is displayed in Figure A.2.4. Design goals of the second simulator included:

- Fast Execution
- Flexible Configuration
- Expandable to handle new Neuron and Synapse Models
- Modular Components
- Continuous time simulation and Discrete time simulation

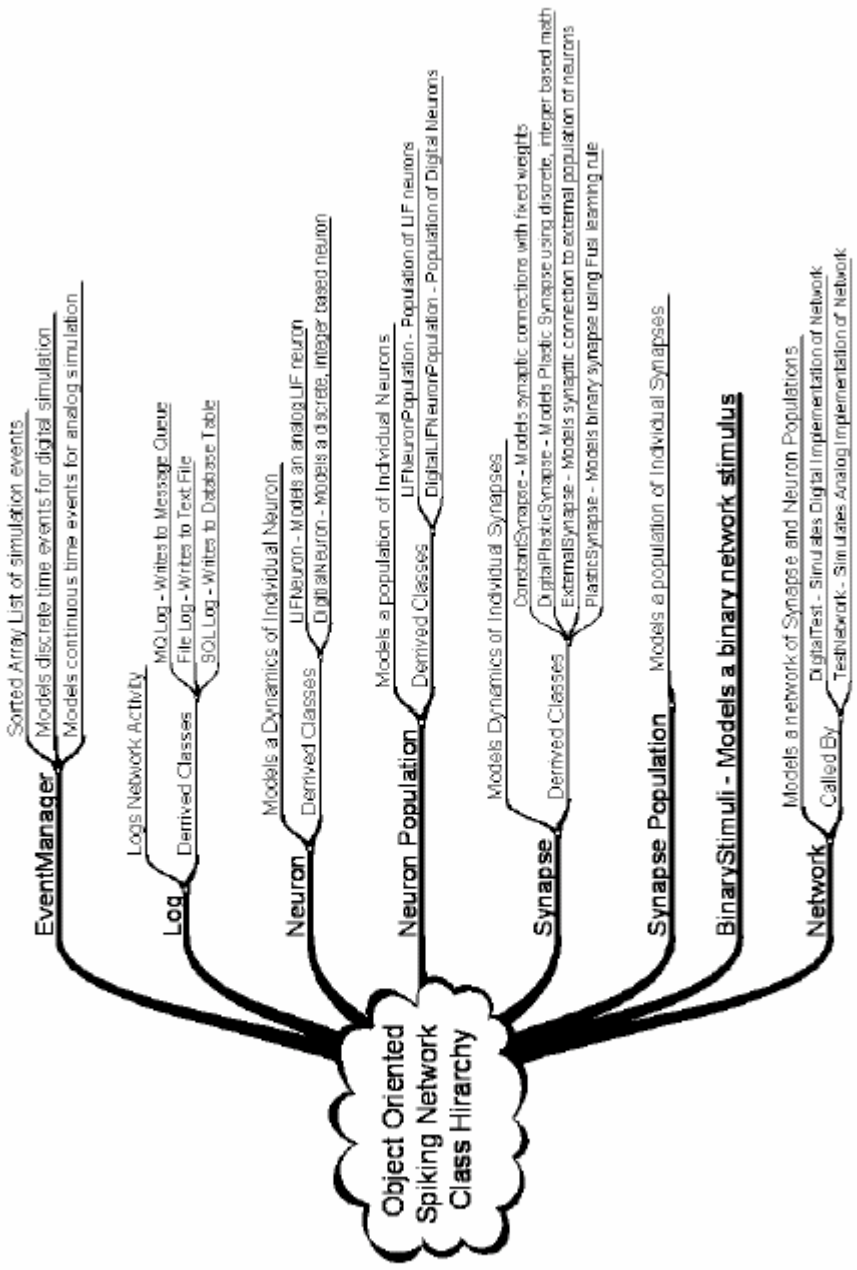


Figure A.2.4: Class Hierarchy of Spiking Network

This tree diagram shows the object oriented structure of the Object Oriented Recurrent Network Simulator. The simulator models analog and digital systems, which may be interchanged. New models of neurons and new learning rules for synapses can easily be created and tested by deriving objects from the base classes.

The object oriented design allows new models of synapses and neurons to be quickly developed for simulation by deriving them from base classes. There is no specified limit to the number of neuron populations, or the combinations of interconnections. For example, an analog neuron population may defined using floating point logic, and then interconnected using digital synapses that use integer logic. This object oriented approach allows a simulated network to combine new neuron models with old neuron models.

Events such as synapse transitions and neuron spikes are logged in a common format. Data may be logged to a *SQL Server* database, stored in a text file, or posted to a message queue.

C# was chosen for the simulation language because it is a rapid prototyping language with the power and elegance of C++[11]. The just-in-time compiler of C# is typically as fast, if not faster, than compiled C++ code, because the just-in-time compilation can perform on-the-fly local optimizations. C# has an extensive class library including items like sorted arrays to make development quick and easy. Additionally, the managed code environment enforced by C# aided in rapid development, because common errors like memory leaks are impossible.

The simulation time of the Object Oriented Recurrent Network Simulator significantly improved over the SQL Recurrent Network Simulator. The simulator could have been further improved, because the optimizations suggested by Reutimann in [7] were not implemented. Simulation speed varies directly with the number of events in the system, so networks with low spike rates simulate faster than networks with high spike rates. On a 1 MHz system with 256 Mb of memory, a simple network of 1500 neurons with a mean spike rate of 32 sps uses about 1.2 minutes per second of simulation. More complex networks with many synaptic transitions use up to 2.6 minutes per second simulated. The Object Oriented Recurrent Network Simulator has a very minimal Graphical User Interface (GUI). The original intention was to develop the simulator, and then add a GUI as time permits. However, the simulator was abandoned before any significant GUI was created. Network parameters are configured directly in code through `test' network objects. Figure A.2.2 is a truncated example of how a network is configured in code.

A.2.3 Experimentation

The Object Oriented Recurrent Network Simulator was first used to test spike-rate stability points. An excel spreadsheet was created to calculate the stabile spike rate, $^{o}(t)$, for a set of network parameters using Equation 1.4. The Object Oriented Recurrent Network Simulator was able to reproduce the results obtained by the SQL Recurrent Network Simulator. It was also able to simulate excitory only networks and functioned according to the theory of Equation 1.4. However, the simulation was less successful using stochastic learning to create bi-stable basins of attraction. After a long period of experimentation, we were unable to balance spike rates and potentiation probabilities in a manner that would successfully create a functional memory. Figure A.2.6 shows representative results, and emphasizes the unstable spike rates.

Experiments with the Object Oriented Recurrent Network Simulator were not a total failure. Although the simulator was eventually abandoned, it still provided the foundation for the final implementation. Two simulators were created to test the dynamics of recurrent spiking networks and their ability to store and recall binary patterns. The SQL Recurrent Network Simulator used SQL server to test the learning algorithm described in Fusi's PhD dissertation[2]. This simulation was successful in demonstrating attractor formation. A second Object Oriented Recurrent Network Simulator was designed using object oriented

techniques that allowed learning rules and neuron logic to be easily modified and tested. The second simulator was used to test pattern storage and recall.

```
network = new Network();
inhibitNeurons = new LIFNeuronPopulation("Inhibit",network);
exciteNeurons = new LIFNeuronPopulation("Excite",network);
inhibitToInhibitSynapses = new ConstantSynapsePopulation
("IISYnapses",network,inhibitNeurons,inhibitNeurons);
inhibitToExciteSynapses = new ConstantSynapsePopulation
("IESynapses",network, inhibitNeurons,exciteNeurons);
exciteToInhibitSynapses = new ConstantSynapsePopulation
("EISynapses",network,exciteNeurons,inhibitNeurons);
exciteToExciteSynapses = new PlasticSynapsePopulation
("EESynapses",network,exciteNeurons,exciteNeurons);
exciteNeurons.NeuronCount = 1200;
exciteNeurons.BackgroundSpikeFactor = 4 * 120;
exciteNeurons.BackgroundSpikeWeight = 25.6;
exciteNeurons.DecayRate = 1024;
exciteNeurons.DepolorizationDelay = .03;
exciteNeurons.InitialVoltageMu = 0;
exciteNeurons.InitialVoltageSigma = 0;
exciteNeurons.Threshold = 1024;
exciteToExciteSynapses.ClockPeriod = 0;
exciteToExciteSynapses.SynapsesPerPostNeuron = 108;
exciteToExciteSynapses.SynapseWeightMuPotentiated = 100;
exciteToExciteSynapses.SynapseWeightMuDepressed = 15;
exciteToExciteSynapses.PercentPotentiated = .1;
exciteToExciteSynapses.SynapseWeightSigma = 0;
```

Figure A.2.5: Truncated Code to Configure a Network

This truncated code snippet demonstrates how a network is configured using object oriented code in the Object Oriented Recurrent Network Simulator. The excitory and inhibitory neuron populations are configured, followed by some synapse populations. Finally, the network is run for 5 s. Note: Not all of the synapse populations are shown here.

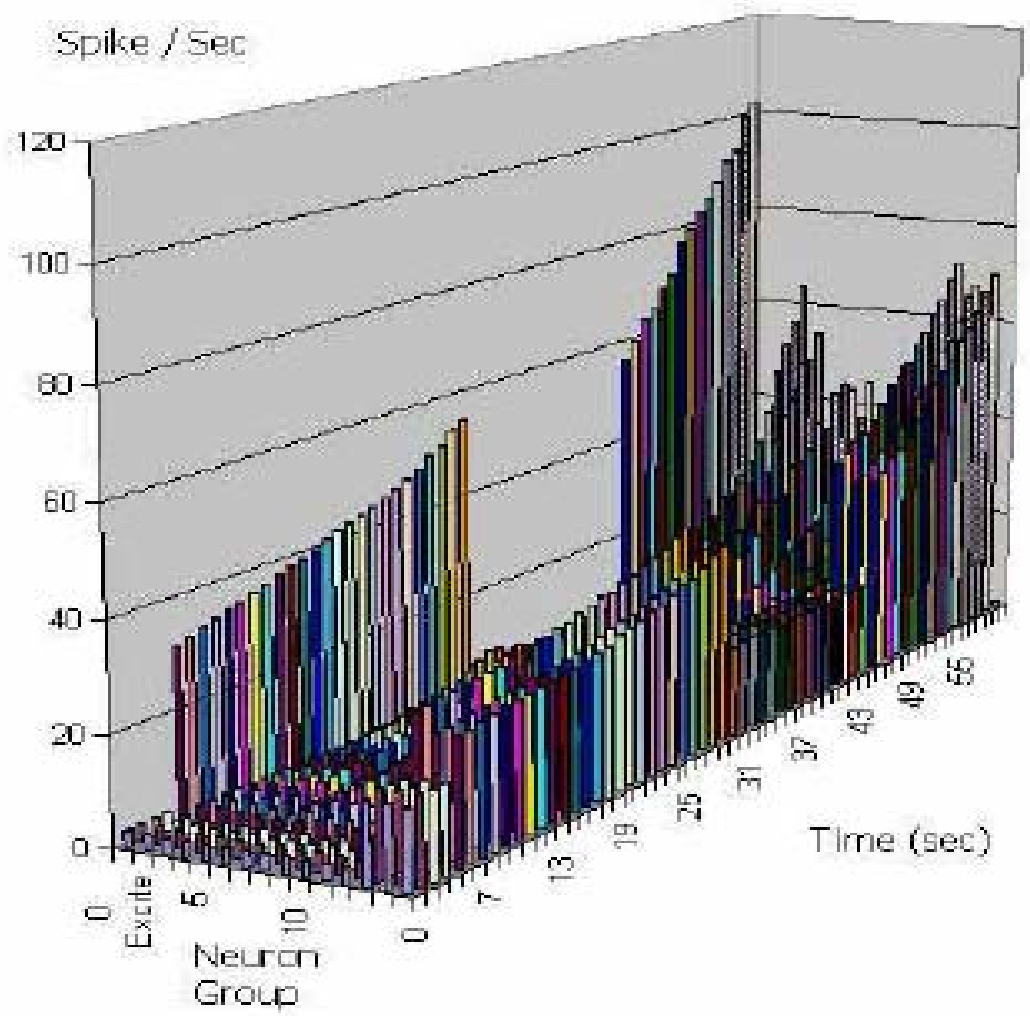


Figure A.2.6: Failed Recurrent Network Test

A stimulus is applied to neurons 1-99 at time 10-30 s. The stimulus is again applied at 50 s. Instead of forming an attractor, the spike rates are unstable and increase indefinitely.

B.1 Protein Memory Modeling

B.1.1. Protein Measurements

In the summer at the Information Directorate, our in-house graduate student, Lem Liou, started work on measurements for protein memory. The idea of using protein as a material for memory storage is made possible by the fact that proteins exist in different conformations, or states, and that in certain proteins these states can be induced by light. Information can be stored in an array of protein clusters arranged and suspended within an optically transparent polymer matrix. By directing light at specific coordinates of the matrix, specific clusters can be switched from "state 1" to "state 2," or better yet is to say, "bit 0" to "bit 1." Bacteriorhodopsin protein has been studied in depth for its potential as a memory storage material, and so far study results are favorable. Experiments were done to confirm the viability of bR as a possible candidate for a novel memory device. More experiments must be done to measure a write efficiency that takes into account diffusive mechanical protein interactions. These measurements must be compared to the theoretical model and discrepancies must be accounted for. An initial model setup was created in JAVA language. This program must be completed with the proper mathematical implementations and eventually built into a full 3-D model. The program is adaptable to add in new variables as new influences to the model behavior are discovered. If necessary, more protein interactions other than mechanical forces can be added into the program. Another key implementation that must eventually be made for the model is the specific time dependent intermediate states of the write and page processes. As new strains of bR are being developed, the cumulative change in the intermediate state properties (intermediate state yields) is what causes the write-efficiency to change. As a future obstacle, these and other implementations to the model should be made.

B.1.2. Background:

The idea of using protein as a material for memory storage is made possible by the fact that proteins exist in different conformations, or states, and that in certain proteins these states can be induced by light. Information can be stored in an array of protein clusters arranged and suspended within an optically transparent polymer matrix [1, 2, 3]. By directing light at specific coordinates of the matrix, specific clusters can be switched from "state 1" to "state 2," or better yet is to say, "bit 0" to "bit 1." Bacteriorhodopsin protein has been studied in depth for its potential as a memory-storage material, and so far study results are favorable.

Bacteriorhodopsin (bR) is a ~26kDa transmembrane protein found in the plasma membrane of *Halobacterium salinarium* [4, 5]. The ~250 monomer residues of bR arrange into seven alpha-helices, folded into a barrel of ~400x400x500 nm dimensions [15]. The photosensitive nature of bR is due to a retinal group being attached to Lys216 [6]. When the chromophore retinal absorbs red light, an *all-trans to 13-cis* isomerization reaction occurs. The 14Carbon-15Carbon bond angle rotates approximately 120 degrees and the 7^{th} alpha-helix (attached to retinal via Lys216) shifts position consequently (figure B.1.1). The 13-cis conformation of bR represents one of five intermediates that occur in the primary (or natural) photocycle of bR which activates upon chromophore contact with ~635nm light. In the primary photocycle (figure B.1.2), ground state bR cycles through the intermediates—K, L, M, N, and sometimes O—then ~8 ms later spontaneously falls back to the original bR resting state. Unfortunately, more often than not, the N state transforms directly back into the original bR state. The total photocycle takes approximately 10 ms. A secondary or branched photocycle (figure B.1.1) occurs when the O-intermediate is activated by a

second exposure to ~690nm light. The branched photocycle consists of the O-intermediate transforming into the P intermediate, which eventually spontaneously transforms into a relatively stable Q-state [1, 2, 3, 7, 8]. The O to P transformation consists of an *all-trans to 9-cis* isomerization reaction of the retinal group, and the P to Q transformation consists of the covalent detachment of the retinal from Lys216. The Q-state is a viable state to be assigned bit 1, while the ground state bR is assigned bit 0. A bit 1 to bit 0 transformation can be achieved only by exposure to high-energy blue light.

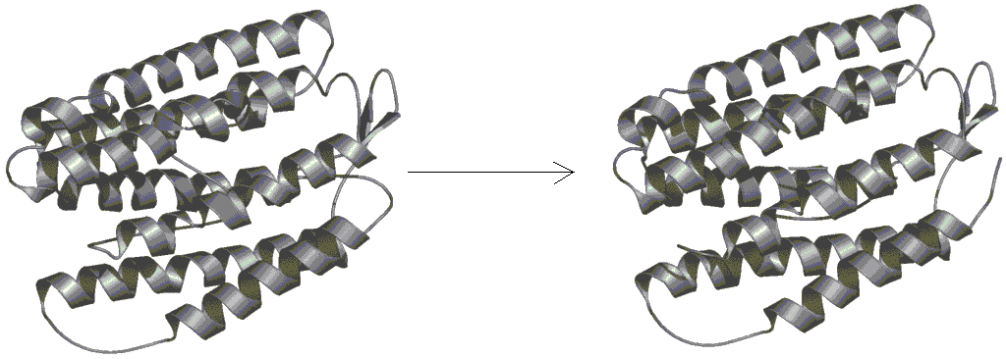


Figure B.1.1 showing a structural representation of two of bR's conformations. *When an isomerization reaction occurs on the retinal, the whole 7th helix shifts position. Therefore a mechanical movement and a structural change is associated with state changing. Pictures taken from [9] and [10].*

B.1.3 Response Times

The response time of bR to light is currently known to be in the order of picoseconds, which has been seen with femtosecond-resolved infrared spectroscopy [11]. The initial isomerization takes place within .5 picoseconds, and after about 3 more picoseconds the K state is reached [11, 8]. The transition from K to L is ~1 microsecond, L to M is ~50 microseconds, M to N is ~1ms, N to O is ~2ms, and O to bR is ~8ms. The relatively slow transition from O back to bR allows time for a second write laser to induce O to P. I can't find any O to P transition rates in the literature. The total time for the normal photocycle is ~10ms. It takes ~2ms to write.

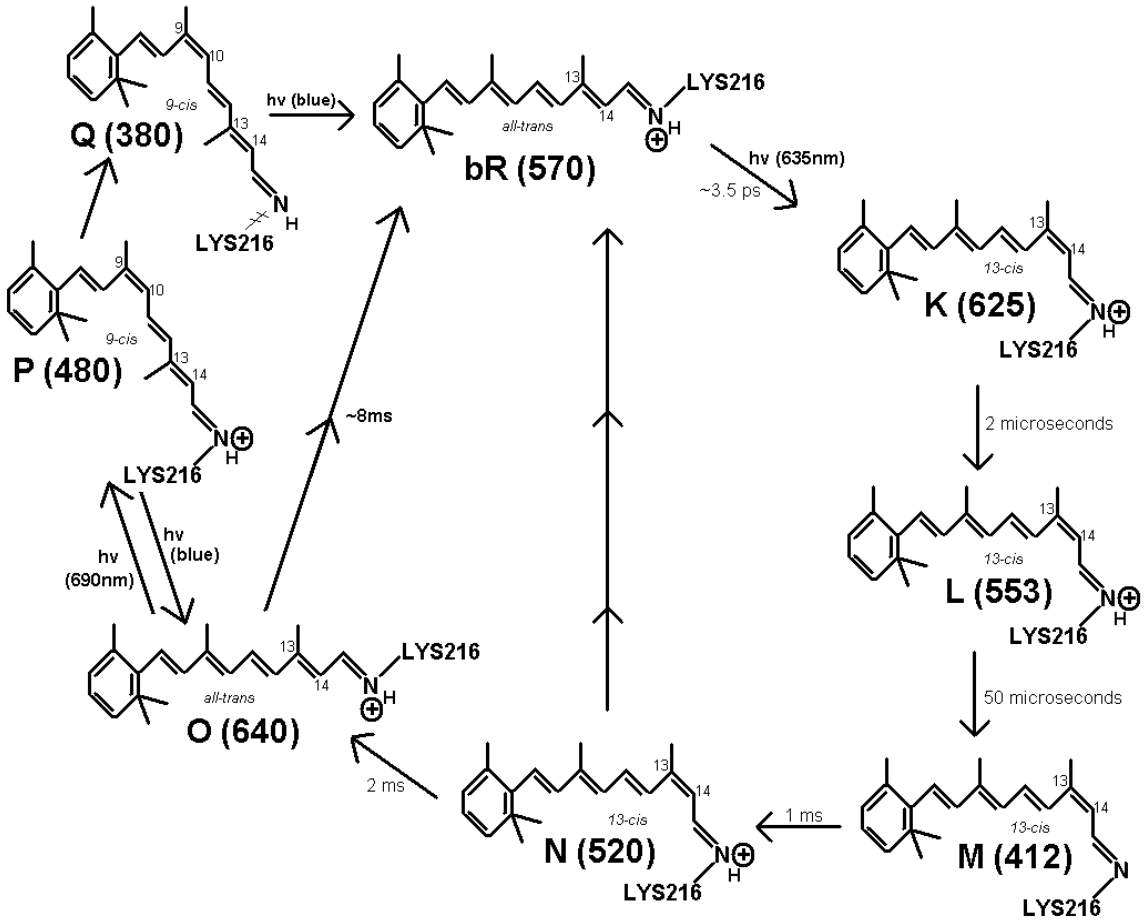


Figure B.1.2 showing the primary (states bR to K, L, M, N, O) and branched (states O to P to Q) photocycles. For advantages of using the P/Q states as the bit 1 state, see [1]. Figure adapted from [12] and [1].

B.1.4 Interface of bR to Bio-memory Devices

Extensive research in the development of bR as a memory-storage device has been and is still being done at Syracuse University [1, 2, 3]. Other researchers of bR-memory devices have looked into using the M-state [12] and K-state [referenced in 1] as the bit 1 state, while other groups such as Syracuse and their collaborators are working on a device using the P and Q states as the bit 1. Their previous research efforts will be used to explain the viability and practical approaches to using bR to store information.

The plasma membrane of *Halobacterium salinarium*, which contains ~80 to 90% wt bR [4, 5], is extracted, purified, sonicated to **fragments** (to be distinguished from **clusters**), and then homogeneously suspended into a polymer hydrogel. The hydrogel is placed in a clear 1x1x3cm cuvette (figure B.1.3). To specify a 3-dimensional coordinate requires two precise laser beams that intersect at the desired coordinate. One laser must have capability to focus in two dimensions, while the other laser has the capability to focus in the 3rd dimension. The first laser, ~635nm, activates the primary photocycle across a whole row of bR-clusters. The activation of the primary photocycle has been coined "paging." A second laser of ~690nm must be shined ~2ms after the first laser, and must be directed down a column which intersects the paged row at the desired coordinate. The

desired coordinate's cluster gets hit with the ~690nm light precisely while it is in the O-state and transforms into the P and Q states, completing the write process. The size of the bit transformed cluster, also known as a **voxel**, is determined by the size of the paging laser and writing lasers' intersection. The paged clusters that were not met by a second ~690nm light reverts back to ground state bit 0, while nothing happens to the column of ground state clusters exposed to the ~690nm light. The read process can be achieved based on absorption properties and yield of the O-state and P/Q states [3]. The erase process is achieved by exposure to blue light.

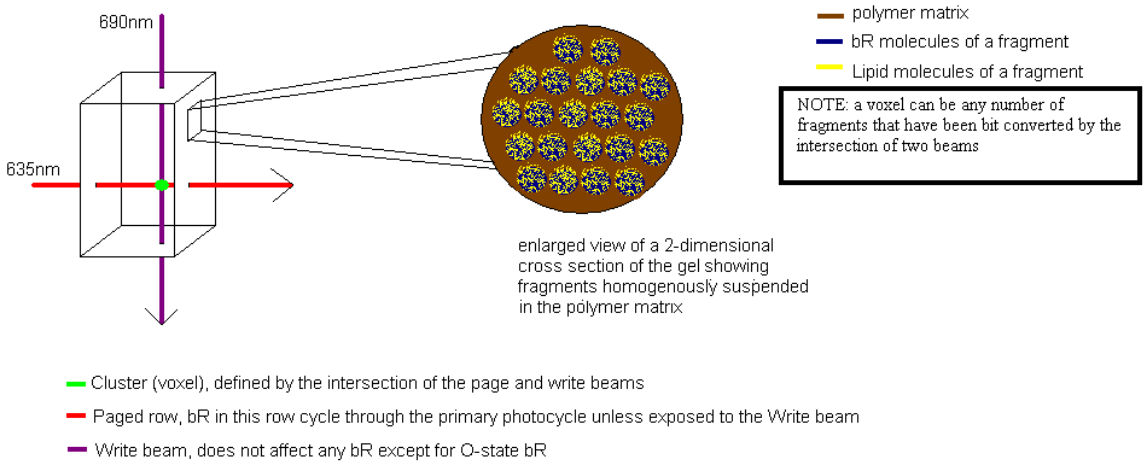


Figure B.1.3 showing the write process for a bR memory gel inside a 1x1x3 cm cuvette and an enlarged view of a cross-section. See text for details. Note the distinction between a voxel (cluster) and a fragment.

B.1.5 Current Status of Bacteriorhodopsin Memory Application

Currently, according to the work from previous contract with Syracuse University, the development of an electronic and computer interface for volumetric memory device prototypes is somewhat complete and fully possible. Implementation of a Fourier-based optical system and implementation of multiplexing techniques, including polarization and gray-scale are not yet realistic because the actual bR memory device has not yet been perfected. One of the biggest obstacles in perfecting the memory-storage device is overcoming the poor quantum efficiency of the write process and to improve reliable state-changing of bR.

In Syracuse's recent experiments it was noted that .21% of a paged bR cluster cycles through the O intermediate, while theoretically it is possible to get a 1.5% yield of O-state [3]. Furthermore, 6.4% of the O state bR, theoretically, convert into the P and Q states. The quantum efficiency of the branched photocycle in the currently developed prototype is anywhere between 1E-4 to 1E-2, while the quantum efficiency of the primary photocycle of bR in nature is ~.67 [5]. Such low quantum efficiency in the memory device increases the likelihood that a read error will occur. When a cluster of bR is paged and then written to, a significant amount of the bR in that cluster must switch states in order to distinguish its bit 1 nature from a nearby unwritten bit 0 cluster. The read process must be able to distinguish a bit 0 cluster from a bit 1 cluster based on the different absorption properties of each state (figure B.1.3), as well as the state change efficiency within each

cluster. Using the O-state to read, as described in [1], helps reduce read-errors because the O-absorption is further away from P and Q absorption than bR absorption (see figure B.1.3). But with still such low O to P quantum efficiency, high intensity beams are necessary to transform a cluster above the read threshold. Such high intensity, however, preclude potential aspects such as multiplexing. High intensity also limits the precision of writing because the voxel size is not small enough. Also described below are speculations of other effects high intensity may have on the memory device's writing efficiency.

The photocyclic properties of the memory storage device must be improved before it can compete with current commercially available memory storage systems. Current attempts at improving quantum efficiencies[1] are underway at several institutions in a somewhat collaborative effort. Genetic engineering of hopeful new bR with more favorable branched photocycle properties is being attempted at University of Connecticut. Chemical modification of the surrounding chemical environment and polymer hydrogel improvements is underway at Carleton University. Other considerations such as optimizing the laser beam characteristics can be studied too. Based on the new problems and challenges facing the development of a competitive bR memory device, in-house experiments on Syracuse-designed memory hydrogels will be done, and a model will be developed to study the energetic aspects of the writing efficiency.

B.2 Approach:

A model of the protein memory device will be created/refined in the Java programming language (selection of Java is primarily due to the high interoperability of the language to multiple end user operating systems). The basic layout of the Java program is shown in figure B.2.1. A data flow diagram is shown in figure B.2.2. The layout represents a large fragment group (fragments being the smallest continuous bR protein units which are then evenly distributed through a non-reactive gel matrix) which is 12x12 in size. In this model the cluster (or voxel, the 3-D area affected by the read/write optical beams) is smaller or equal to the size of the fragment. The items in the 2-D array represent photoactive molecules of the memory device (bR in our situation). The user can enter values into a text box and set the value of an independent variable condition to the text box value with the drop down menu. For instance, beam area can be set at any value up to 12 molecules (which would set voxel size or the effective diffusion-constrained area to 12x12 molecules large).

[1] Quantum Efficiency is defined as the ratio between the photons used for a chemical reaction and the total absorbed photons. Quantum Efficiency is sometimes interchanged with the term Quantum Yield.

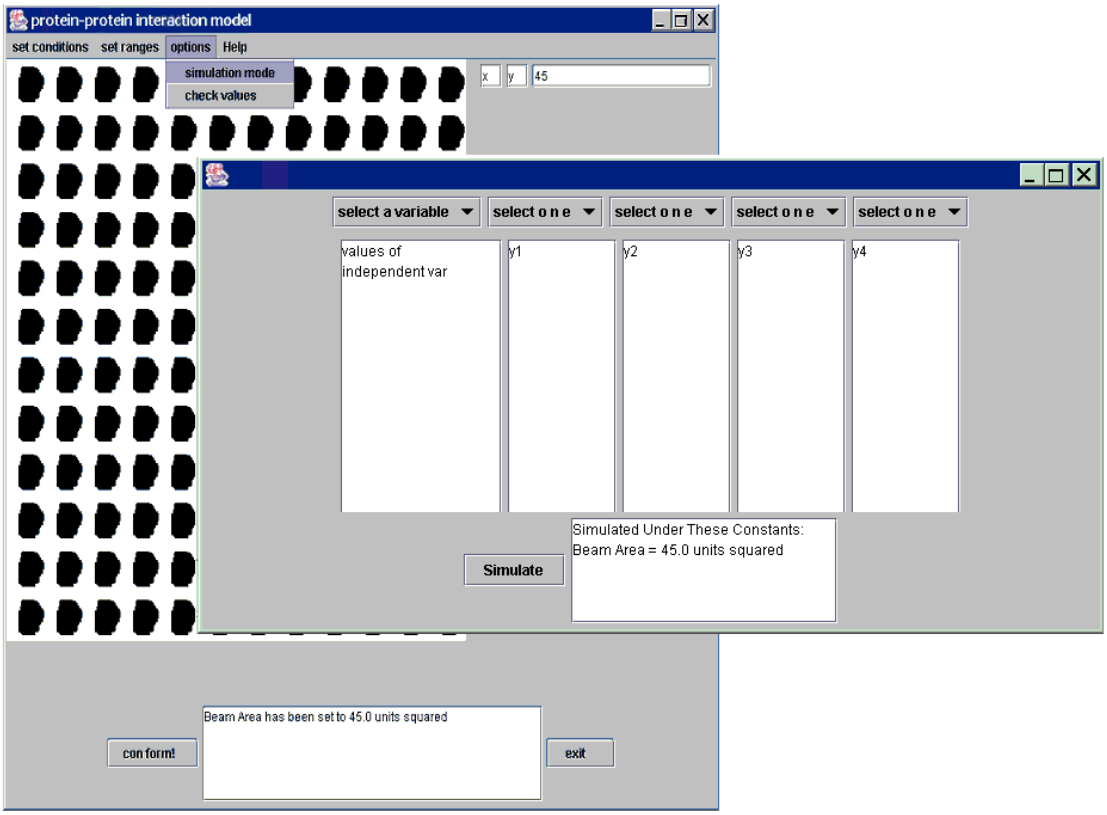


Figure B.2.1. Shows the layout of the JAVA graphical user interface (GUI) program. The pop up GUI shown is the simulation mode

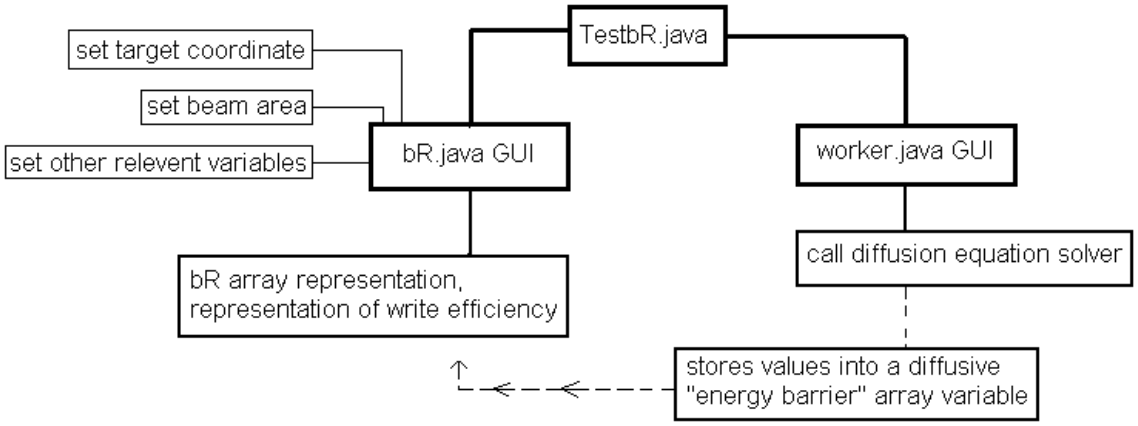


Figure B.2.2. Shows a (proposed) data flow diagram for the JAVA program.

In simulation mode, the diffusion equation will be applied to the system based on the inputted quantum yields and laser beam energies. The program will calculate the total mechanical energy reaching each bR molecule in the voxel due to mechanical diffusion of energy from each photoexcited bR molecule. The diffused energy will be accounted for in the energy barrier of the writing operation. Also in the simulation mode, values from dependent variables can be listed in correspondence to varying manipulated variables.

Dependent lists can show values such as the energy reaching a specific bR in the voxel (figure B.2.3). In addition, the resulting write-efficiencies obtained will be graphically represented in the 2-D array showing which bR molecules are likely to have undergone conformation upon exposure to the write laser (figure B.2.4). The bR that get conformed will be mathematically based on the beam area and relative coordinates chosen in the text boxes, and depend on the diffusion energy involved. The "simulate" JButton needs to call a solving program that will solve the diffusion equation for each bR array member within the beam area (set in the main layout) and return each value to a corresponding "energy barrier" array variable. For each bR molecule in the coordinate array, the "conform" JButton will evaluate the following statement for each bR molecule within the set voxel perimeter:

IF (write barrier) + Σ(energy from diffusion equation)/(Avogadro's number) < (intensity/1000)*(duration/1000), THEN bit conformation = true

A major goal of the modeling/simulation work is to gain additional insight into the write efficiencies of the bR material (a main issue preventing widespread adoption of optical protein memory remains low write efficiencies, currently on the order of 2-4%). By taking protein interactions into account (based on mechanical energy diffusions) we seek to discover new methods of protein-optimization whether by new methods of sample preparation, modifications to the optical recording device, or in the directed evolution of the protein itself in order to perfect the protein memory device.

Building on work previously conducted at Syracuse University (under AFRL contract), the in-house effort also seeks to explore new methods and in particular new materials that may be viable substitutes for the bR protein technology (while bR protein optimization research via genetic engineering is ongoing at multiple research sites, it remains a bottleneck of the commercial viability of bR as a memory material). New macromolecules have recently been identified for prospective optical memory device applications and a portion of the effort will be spent investigating their potential as memory substrates and as participants in bio-inspired cognitive computing paradigms.

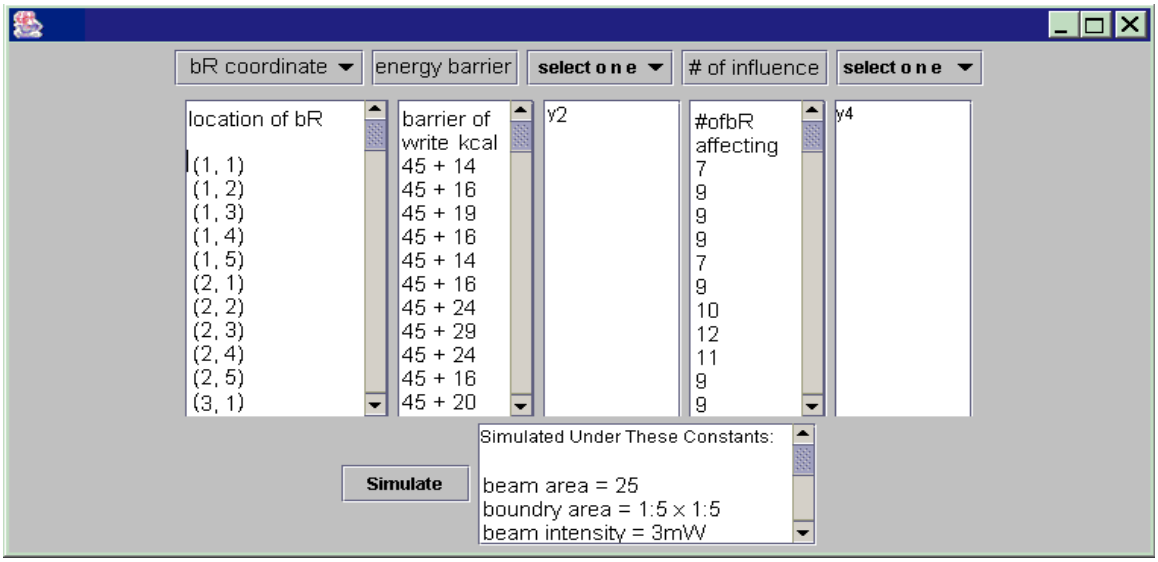


Figure B.2.3 shows how the simulation box will function. This program does not yet have mathematical implementations and therefore does not yet have the capability to calculate the hypothetical values seen in this box. The energy barrier values will be converted from kcal/mol to joules/atom before computation begins, for dimensional compatibility.

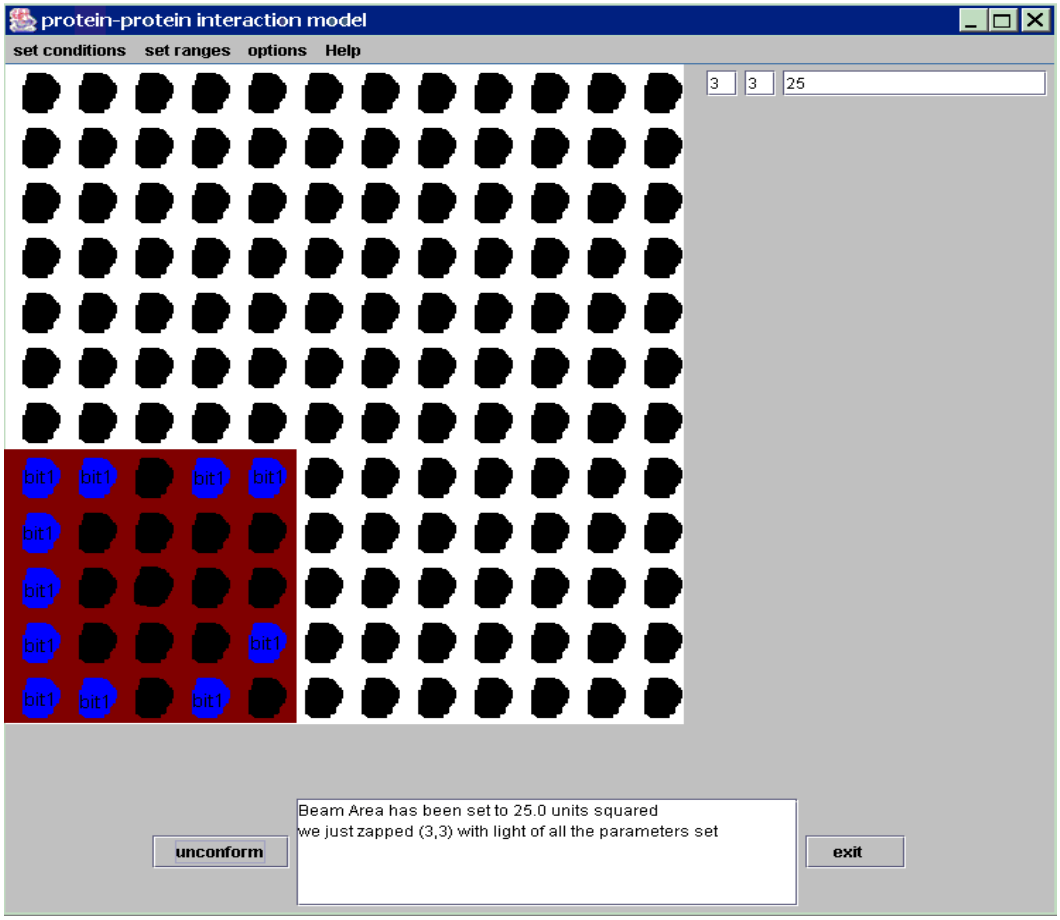


Figure B.2.4 shows how the 2-D molecule layout will function.

B.2.1 Protein Modeling

The emphasis is on developing mathematical models suitable for computer simulation. A system of proteins arranged into a one–dimensional matrix, containing ion particles has been developed. The proteins are considered fixed in their positions together with their surrounding membranes. The ions move between proteins and may also be captured/released by proteins. The presently modeled mechanisms of ion movement are ***diffusion*** and ***active membrane transport***; additional mechanisms are under investigation.

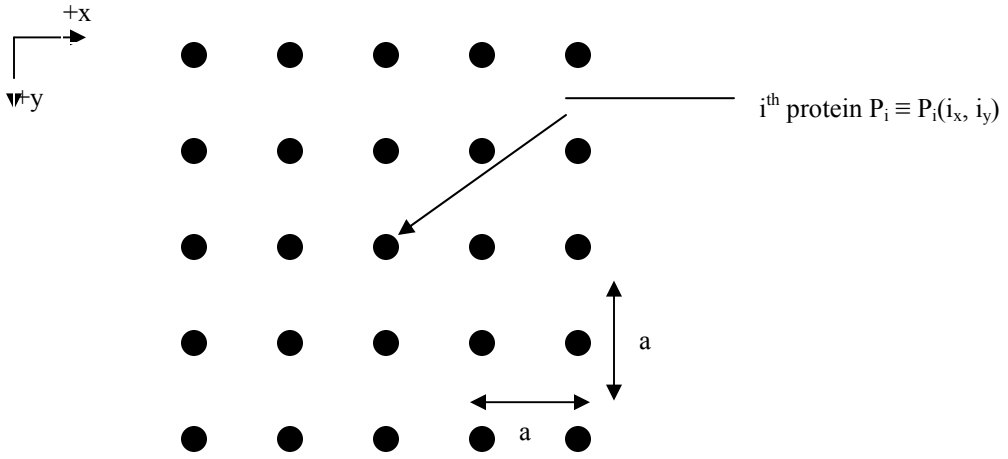
B.2.2 Two dimensional protein matrix model

The matrix model, builds upon a model of a two – dimensional system of proteins. Two physical forces (potential difference and diffusion) are developed and employed to induce time – dependent changes in two fluxes (electric current and ion mass density) flowing within a matrix of proteins with fixed positions. The forces and fluxes are coupled using the Onsager Reciprocal Relation.

From a physical viewpoint, each protein is viewed as an electrode with a time – varying potential; ions and current then flow between neighboring electrodes, so that ion concentration also changes with time.

From a "logic circuit" viewpoint, as defined and modeled by Hennie [20], each protein is viewed as logical cell in an array of identical cells, each receiving primary input signals from the outside world and intercellular inputs from nearest neighbors.

The protein matrix is a fixed, regular two dimensional array of individual proteins, i.e. a is 'a' constant at all times:



The forces and fluxes are defined as follows:

$J_1 \equiv$ current, i.e. the usual conduction of electrons,
$J_2 \equiv$ ion flux, i.e. the motion of mass particles,
$X_1 \equiv$ electromotive force, i.e. a voltage difference,
$X_2 \equiv$ concentration gradient of ions.

Following the paper by Onsager [16], the coupled reciprocal relations are:

$$X_1 = R_{11}J_1 + R_{12}J_2$$
$$X_2 = R_{21}J_1 + R_{22}J_2 \qquad (1)$$

Here R_{11} is the usual resistance of Ohm's law, and R_{22} is a diffusion coefficient. The above equations can be rewritten in matrix form:

$$\vec{X} = \overline{R}\vec{J} \quad , \quad \text{or} \quad \vec{J} = \overline{R^{-1}}\vec{X}$$

with

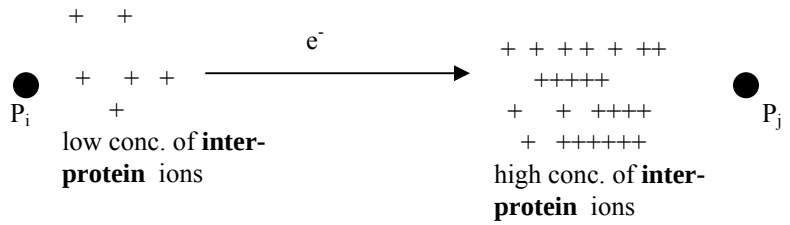
$$\overline{R}^{-1} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \equiv \overline{L} \qquad \text{i.e.}$$

$$J_1 = L_{11}X_1 + L_{12}X_2$$
$$J_2 = L_{21}X_1 + L_{22}X_2 \qquad (2)$$

Physical interpretation of each term of (2).

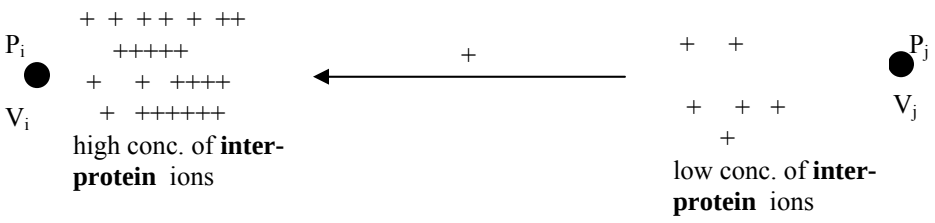
$L_{11}X_1$. L_{11} is conductance, X_1 is a voltage drop $\Rightarrow$ this term is the usual Ohm's Law for conduction of electric current by electrons. The units are [amp] = [mho][V].

$L_{12}X_2$.



The nonzero concentration gradient ($X_2 \neq 0$) induces a flow of electrons. If the units of X_2 are [mass (unit area)$^{-1}$ (unit distance)$^{-1}$] then the units of L_{12} are [amp area distance (unit mass)$^{-1}$].

$L_{21}X_1$.

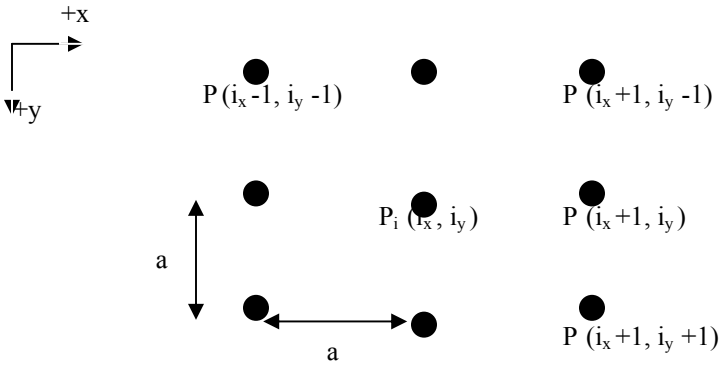


Potential difference $X_1 \equiv V_i - V_j$ between P_i and P_j, which act as electrodes, induces flow of interprotein ions. In the case pictured here, X_1 causes a flow against the concentration

gradient. The units of J_2 are [mass (unit time)$^{-1}$], the units of X_1 are [V] $\Rightarrow$ the units of L_{21} are [mass (unit time)$^{-1}$(unit voltage)$^{-1}$]. (The electric current caused by this flow of ions is $nq\,J_2$, q = unit charge, n = # of charges per ion).

$L_{22}X_2$. This term describes the usual flow of particulate matter under the influence of a concentration gradient. The units work out as [mass (unit time)$^{-1}$] = [area distance (unit time)$^{-1}$] [mass (unit area)$^{-1}$ (unit distance)$^{-1}$].

Modeling the "forces" $X_{1,2}$.
At present, only nearest neighbor interactions are considered. For protein P_i :



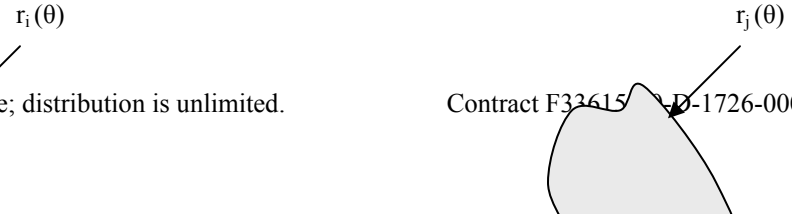
To model the potential difference X_1 between two neighboring proteins, first model the charge density ρ around each protein. For example, for protein P_i :

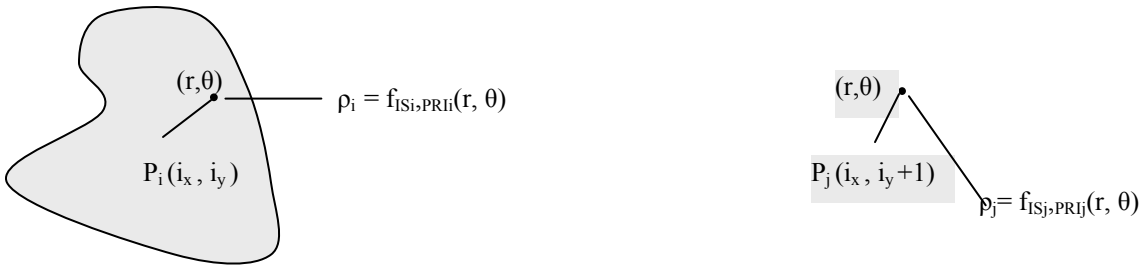
$$\rho_i \equiv \rho_i(IS_i\ ,\ PRI_i\ ,\ r_i(\theta),\ r,\ \theta) \qquad [\text{charge (unit area)}^{-1}\ ]$$

where

IS_i ≡ present internal state of P_i (e.g. one state of the bacteriorhodopsin photocycle),
PRI_i ≡ primary input to P_i (e.g. photons from the outside world),
$r_i(\theta)$ ≡ bounding curve surrounding the charge distribution ρ_i of P_i ,
(r, θ) ≡ polar coordinate points on the interior of $r_i(\theta)$, $0 < r < r_i$, $0 < \theta < 2\pi$.

IS_i and PRI_i can affect the distribution ρ_i since conformational changes and external inputs may have effects on the spatial configuration of the charges held by P_i . E.g. a change in a protein's tertiary structure may shield or expose some of its charge; a photon striking P_i may induce changes in structure or affect charge distribution. ρ_i is also affected by ions that P_i may capture or release.





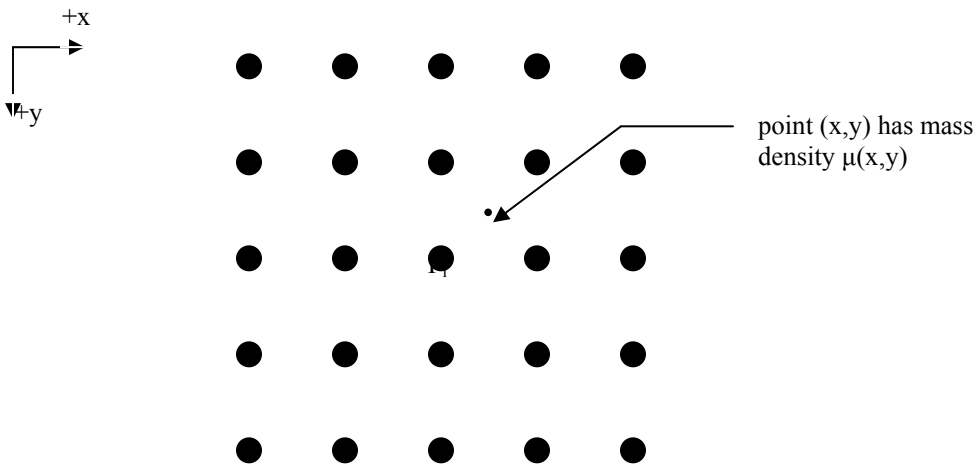
The potentials at the points $P_{i,j}$ (nominally at the "centers" of proteins $P_{i,j}$) are formally calculated as:

$$V_{i,j} = \int_0^{2\pi} \int_0^{r_{i,j}} \frac{\rho_{i,j}(r,\theta)}{r} r dr d\theta$$

Then:

$X_1 = V_i - V_j$ for the potential difference between P_i and P_j .

To model the force X_2 , which is driven by the ion concentration gradient, the mass density of ions, μ(x,y), must first be modeled:



If point (x,y) falls within the boundary $r_i(\theta)$ of some P_i , then the mass density of ions μ(x,y) is considered "captured" by P_i , and μ can contribute a charge density of up to nqμ(x,y) to the total charge density $\rho_i(x,y)$ at point (x,y); q is the fundamental charge and n is the number of such charges per ion. If (x,y) does not fall within any $r_i(\theta)$, then ions are free to flow through this point, under the influences of X_1 , X_2 . X_1 is modeled above.

X_2 is modeled as follows. The concentration gradient of ions is given by **grad** μ, so that the ion flux due to this gradient is D_m **grad** μ, where D_m is the mass diffusion coefficient. D_m **grad** μ is the quantity $L_{22}X_2$ of equation (2), with L_{22} as D_m and X_2 as **grad** μ.

Since the ions can move, μ is clearly also a function of time. The "full space – time" behavior of μ can be modeled using Fick's Second Law of Diffusion:

$$\frac{\partial \mu}{\partial t} = \nabla \bullet (D_m \nabla \mu)$$

Since, as discussed above, μ can contribute to ρ, ρ is also a function of time. A changing ρ may induce a change in the protein's internal state IS, causing further changes in ρ. A change in ρ may cause a change in the protein's potential, so that X_1 is generally also a function of time; clearly X_2 varies with time, and may be affected by a protein's ability to capture or release ions at any given instant. Thus the coupled equations (2) are time dependent, and define an evolving system under the influence of two forces $X_{1,2}$ and two "messengers" $J_{1,2}$. One or more equilibrium states must be defined for the protein matrix, i.e. in the absence of all external inputs (the PRI primary inputs of the model) $J_1 = J_2 = 0$, that is, no messages are being exchanged between any proteins. The subsequent application of PRI signals (e.g. photons) causes a disturbance in the system, which undergoes a series of state changes in space and time, ostensibly reaching equilibrium (the same or a different one)again.

C.1 Analog Re-Configurable Computational/Processing Units:

Reusable Analog Integrated Circuit Components are needed for hardware signal processing and for System-on-A-Chip (SoC) Integration. These include: Microwave Operational Amplifier(s); Switched Capacitor Filters; and Voltage References. During this research period, our efforts have been focused on re-configurable microwave Op-Amp. A prototype 0.18 CMOS SOI design structure fort RF high-gain op-amp was submitted for prototype implementation through the MIT Lincoln lab, FDSOI process. The design was submitted, but has not been received by Oakland University to test results as of the date of the final report.

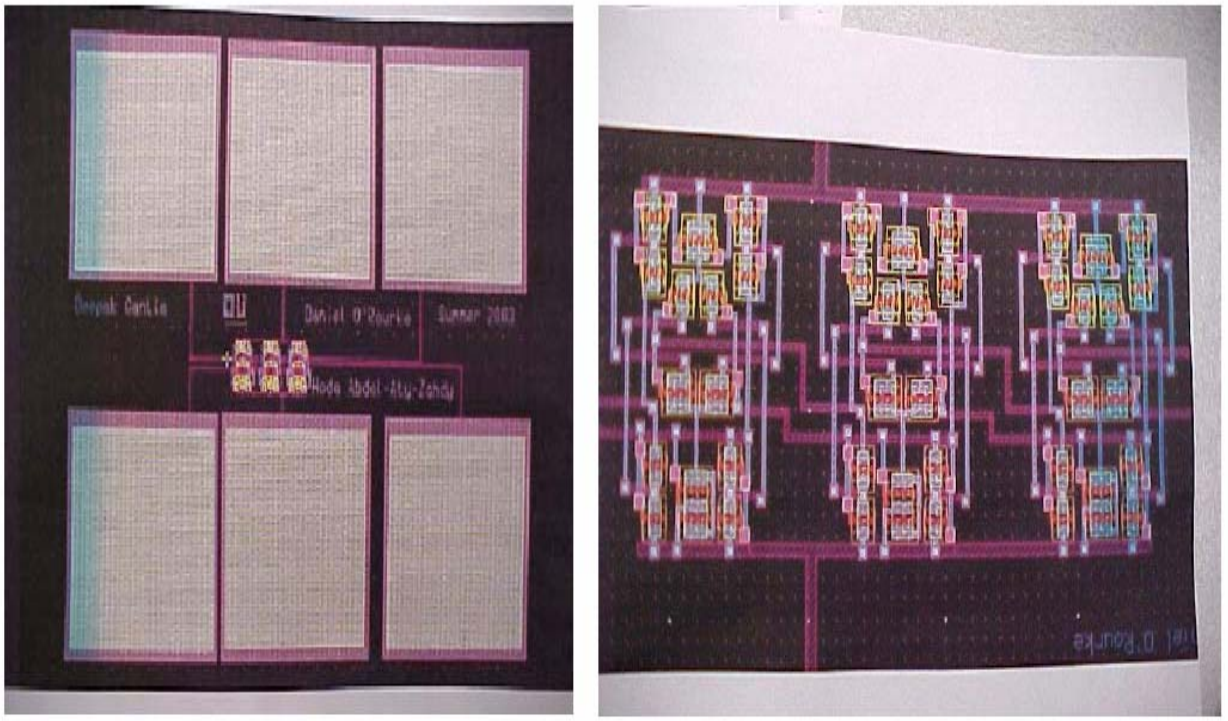


Figure C.1 High-gain RF Op-Amp on 0.18 um FDSOI CMOS Processed at the MIT-LL

References

Section A

1. L. Buck, The molecular architecture of odor and pheromone sensing in mammals," *Cell*, pp. 611{618, Mar. 1996.
2. S. Fusi, Neural networks: Function and learning in theory and experiment,"Ph.D. dissertation, Jerusalem University, Dept. of Physics, 1999.
3. S. Fusi, P. D. Giudice, and D. Amit, Neurophysiology of a vlsi spiking neural network," in *Proc. of the IEEE-INNS-ENNS Intl. Joint Conf. on Neural Networks*, 2000, pp. 121-126.
4. M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*. PWS Publishing Company, 1995.
5. C. Peterson, R. Malenka, R. Nicoll, and J. Hopfield, All-or-none potentiation at ca3-ca1 synapses," *Proc. Natl. Acad. Sci, USA*, vol. 95, pp. 4732-4737, Apr. 1998.
6. D. Amit and S. Fusi, Learning in neural networks with material synapses," *Neural Computation*, vol. 6, pp. 957-982, 1994.
7. J. Reutimann, M. Giugliano, and S. Fusi, Event-driven simulation of spiking neurons embedded in very large networks," in *Proc. of the World Congress on Neuroinformatincs (ARGESIM/ASIM)*, 2001.
8. S. Fusi, M. Annunziato, D. Badoni, A. Salamon, and D. Amit, Spike-drive synaptic plasticity: Theory, simulation, vlsi implementation," *Neural Computation*, vol. 12, pp. 2227-2258, 2000.
9. N. Brunel, F. Carusi, and S. Fusi, Slow stochastic hebbian learning of classes of stimuli in a recurrent neural network," *Network*, vol. 9, pp. 123-152, Feb.1998.
10. M. Mattia and P. D. Giudice, \E±cient simulation of large networks of spiking neurons and dynamical synapses," *Neural Computation*, vol. 12, pp. 2305-2329, 2000.
11. S. Lippman, *C# Primer*. Developmentor, 2002.

Section B

1. Birge et al. *Biomolecular Electronics: Protein-Based Associative Processors and Volumetric Memories*. Journal of Physical Chemistry B, 1999. **103**:**49,** p.10746-10766.
2. Stuart, J., Marcy, D., Wise, K., Birge, R. *Volumetric Optical Memory Based on Bacteriorhodopsin*. Synthetic Metals, 2002. **127**, p.3-15.
3. Stuart, J. *Protein-Based Three-Dimensional Memories*. AFRL-IF-RS-TR-2001-279 Final Technical Report, January 2002. Air Force Research Laboratory, Information Directorate, Rome Research Site, Rome, New York.
4. Tajkhorshid, E., Baudry, J., Phillips, J., Hayashi, S. Bacteriorhodopsin and the Purple Membrane. 7 April 2003. Theoretical and Computational Biophysics Group, University of Illinois Urbana-Champaign. 9 June 2003. http://www.ks.uiuc.edu/Research/newbr/.
5. Baudry, J., Tajkhorshid, E., Molnar, F., Phillips, J., and Schulten, Klaus. *Molecular Dynamics Study of Bacteriorhodopsin and the Purple Membrane*. Journal of Physical Chemistry B, 2001. **105**:**5**, p.905-918.
6. Tajkhorshid, E., Baudry, J., Schulten, K., Suhai, S. *Molecular Dynamics Study of the Nature and Origin of Retinal's Twisted Structure in Bacteriorhodopsin*. Biophysical Journal, 2000. **78**, p.683-693.
7. Popp et al. *Photochemical Conversion of the O-intermediate to 9-cis-Retinal-Containing Products in Bacteriorhodopsin Films.* Biophysical Journal, 1993. **65**, p. 1449-1459.
8. Tallent et al. *Photochemistry in Dried Polymer Films Incorporating the Deionized Blue Membrane Form of Bacteriorhodopsin*. Biophysical Journal, 1998. **75**, p. 1619-1634.
9. **MolMovDB: analysis and visualization of conformational change and structural flexibility.** N Echols, D Milburn, M Gerstein (2003) *Nucleic Acids Res* 31: 478-82.
10. **The morph server: a standardized system for analyzing and visualizing macromolecular motions in a database framework.** WG Krebs, M Gerstein (2000) *Nucleic Acids Res* 28: 1665-75.
11. Herbst, J., Heyne, K., Diller, R. *Femtosecond Infrared Spectroscopy of Bacteriorhodopsin Chromophore Isomerization*. Science, 2002. **297**, p.822-825.
12. Tokes, Sz., Orzo, L., Varo, Gy., Roska, T. *Bacteriorhodopsin as an Analog Holographic Memory for Joint Fourier Implementation of CNN Computers*. Research Report DNS-2000. Analogical and Neural Computing Research Laboratory, Computer and Automation Institute, Budapest, April 2000.
13. Bechinger et al. *Refinement of the Geometry of the Retinal Binding Pocket in Dark-Adapted Bacteriorhodopsin by Heteronuclear Solid-State NMR Distance Measurements.* Biochemistry, 2000. **39**, p. 10066-10071.
14. Kuczera, K., Hermone, A. *Free-Energy Simulations of the Retinal Cis → Trans Isomerization in Bacteriorhodopsin.* Biochemistry, 1998. **37**, p. 2843-2853.
15. Stryer, L., Berg, J., Tymoczko, J. Biochemistry. 5th edition. W.H. Freeman and co. New York, NY. p. 330-331.
16. L. Onsager, "Reciprocal Relations in Irreversible Processes I," *Physical Review*, vol. 37, pp. 405-26, Feb. 15,1931.
17. S. Frickenhaus, R. Heinrich, "Kinetic and Thermodynamic Aspects of Lipid Translocation in Biological Membranes", *Biophysical Journal*, vol. 76, pp.1293-1309, Mar. 1999.
18. Griffiths, "Introduction to Electrodynamics, 2nd ed.," Prentice Hall,1989.
19. R. Pierret, "Semiconductor Fundamentals, Volume I, 2nd ed.," Addison-Wesley, 1988
20. F. Hennie III, "Iterative Arrays of Logical Circuits," MIT Press, 1961.

APPENDIX-I

Exploring the BioComputing Frontier

In collaboration with: *Dr. R.L. Ewing (WPAFB/IFTA)*
Mr. L. Liou, (Summer Student 2003),and
Mr. J.B. Moncrief (WPAFB/IFTA)

ABSTRACT

Stop thinking that the universe is best described by Boolean order operations, consider the world of biocomputing. The idea in exploring the multidimensional characteristics of biocomputing systems involves multilevel *protein* logic interaction with hybrid digital systems (bio/digital/analog systems). In the *protein* biocomputing world, all models are multilevel and nonlinear exponential in nature. These biocomputing systems are linearized by boundary limited frequency, time response and dimensional ranges. The protein model and its *correct* development is probably the most difficult aspect for biocomputing issues. Previous protein models were not dealing with 2 and 3 dimensional models, while the current protein models are now faced with high frequency design, due to the emergence of bio-nanodevices. These future biocomputing devices will be chemical sensors, fluidic computer systems, optical switches, nanoactuators, self-repairing polymorphic computing systems, cognitive processors and biorobotics.

I. Introduction

Engineering terminology is changing with many new areas appearing. As such was the case a century ago when electrical engineering was a course offered in civil engineering, biotechnology and nanotechnology are the leading edge *new* emerging disciplines for bio-inspired electronics and the tools for development of protein logic. The temporal and spatial domains of protein logic, as shown in Fig. 1, represents the biological and chemical characteristics compared to CMOS, but in the 2 and 3 dimensional protein models, phenomena's such as electrical, magnetic, mechanical, optical, and thermal properties are included for performance specifications issues. Desired specifications at the system level are: sensitivity, measurand range, stability, resolution-accuracy, speed of response, temperature variations, and power characteristics. In the early development of this technology, the envisioned biocomputing system will contain a digital output stage, wherein multilevel protein signals are converted to logic, and classified as a protein logic system. The envisioned protein logic systems will be hybrid in nature, being composed of both organic and inorganic components. The biocomputing architecture involves a digital control system that utilizes the protein logic system as a cognitive processing device.

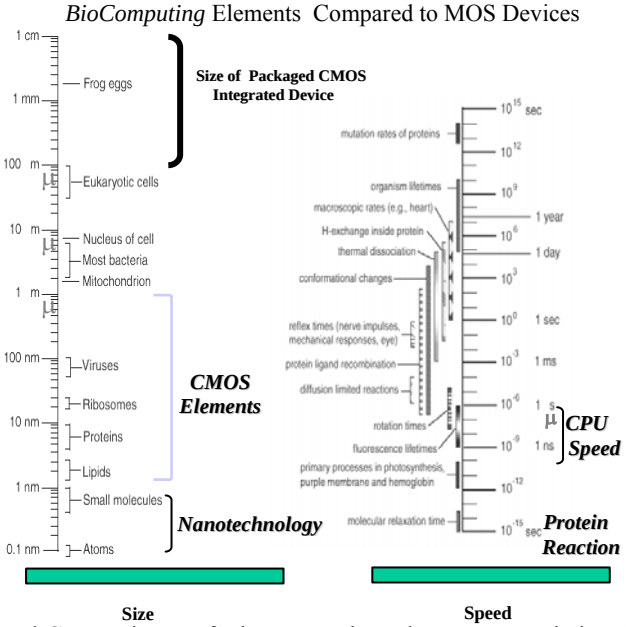


Figure 1. Size and Speed Comparisons of Bio-computing Elements to Existing CMOS Digital Technology.

Performance specifications requires the protein logic memory system to store an optical input despite various system disturbances. Satisfactory information storage with disturbances is associated with a hybrid digital control disturbance rejection system evolved around the multilevel protein logic and its various states.

II. Bio-Memory

Proteins react to vapor and gas concentrations, so the question becomes: Can they react and store information and can they provide a polymorphous memory structure for contextual applications? To answer these questions, several biological molecules are under consideration for use in computer polymorphous hardware, but the bacterial protein bacteriorhodopsin has generated the most interest. During the past 12 years, parallel-processing devices, with three-dimensional data storage hardware (Fig. 2) and recently neural networks based on this protein logic, have been built by the Air Force Research Laboratory's Information Directorate. Interest in bacterial-rhodopsin dates back to the early 1970s, when Walther Stoeckenius of the University of California at San Francisco and Dieter Oesterhelt, now at the Max Planck Institute for Biochemistry in Martinsried, discovered that the protein exhibited unusual properties when it was exposed to light. Found in the membrane of Halobacterium salinarium, bacteriorhodopsin enables the bacterium to grow when the concentration of oxygen is insufficient to otherwise sustain the organism. When struck by light, the protein changes its structure and transports a proton across the membrane, thereby supplying energy to maintain cell metabolism. Microfiche films, which are self replicating DNA (gene logic), called Biochrome, are composed of bacteriorhodopsin. Both rhodopsin and bacteriorhodopsin are complex proteins that include a light-absorbing component known as a chromophore.

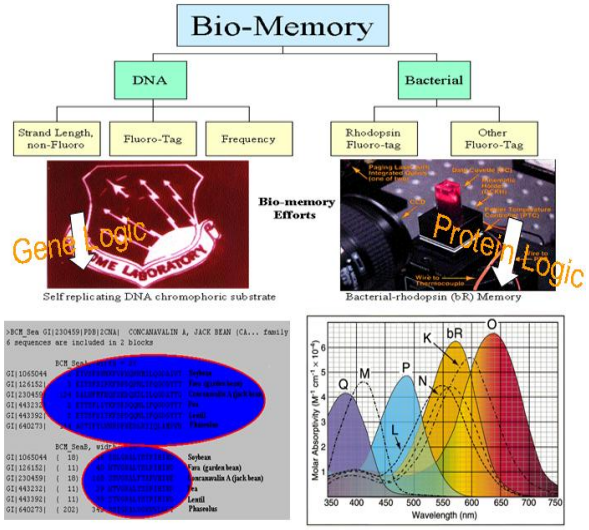


Figure 2. BioMemory Polymorphic Technology of Gene and Protein Logic.

The chromophore absorbs energy from light, triggering a complex series of internal motions that result in dramatic changes in the structure of the larger protein. These changes alter the protein's optical and electrical characteristics. (In protein logic, the resting state '0' is known as bR, and each intermediate in the series is identified by a letter of the alphabet, as shown in Fig.2). The various intermediates can be used to represent bits of data. Moreover, the intermediates absorb light in different regions of the spectrum.

The idea of using protein as a material for memory storage is made possible by the fact that proteins exist in different conformations, or states, and that in certain proteins these states can be induced by light. Information can be stored in an array of protein clusters arranged and suspended within an optically transparent polymer matrix [1, 2, 3]. By directing light at specific coordinates of the matrix, specific clusters can be switched from "state 1" to "state 2," or better yet is to say, "bit 0" to "bit 1." The Q-state is a viable state to be assigned bit 1, while the ground state bR is assigned bit 0. A bit 1 to bit 0 transformation can be achieved only by exposure to high-energy blue light.

III. Pico-second Response Time for Bio-memory

The response time of bR to light is currently known to be in the order of picoseconds, which has been seen with femto-second-resolved infrared spectroscopy [4]. The initial isomerization takes place within 0.5 picoseconds, and after about 3 more picoseconds the K state is reached. The transition from K to L is ~1 microsecond, L to M is ~50 microseconds, M to N is ~1ms, N to O is ~2ms, and O to bR is ~8ms. The relatively slow transition from O back to

bR allows time for a second write laser to induce O to P. The total time for the normal photocycle is ~10ms and ~2ms to write.

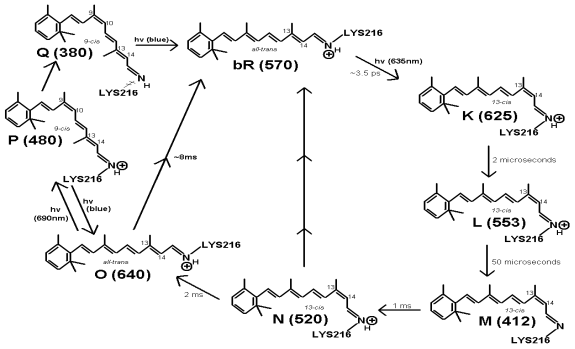


Figure 3. Showing the primary (states bR to K, L, M, N, O) and branched (states O to P to Q) photocycles.

Extensive research in the development of bR as a memory-storage device has been and is still being done. To specify a 3-dimensional coordinate requires two precise laser beams that intersect at the desired coordinate. One laser must have capability to focus in two dimensions, while the other laser has the capability to focus in the 3rd dimension. The first laser, ~635nm, activates the primary photocycle across a whole row of bR-clusters. The activation of the primary photocycle has been coined "paging." A second laser of ~690nm must be shined ~2ms after the first laser, and must be directed down a column which intersects the paged row at the desired coordinate. The desired coordinate's cluster gets hit with the ~690nm light precisely while it is in the O-state and transforms into the P and Q states, completing the write process. The size of the bit transformed cluster, also known as a **voxel**, is determined by the size of the paging laser and writing lasers' intersection. The paged clusters that were not met by a second ~690nm light reverts back to ground state bit 0, while nothing happens to the column of ground state clusters exposed to the ~690nm light. The read process can be achieved based on absorption properties and yield of the O-state and P/Q states [3]. The erase process is achieved by exposure to blue light.

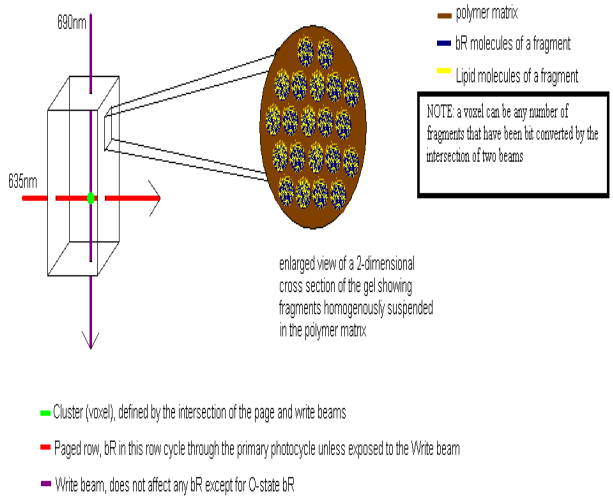


Figure 4. Showing the write process for a bR memory gel inside a 1x1x3 cm cuvette and an enlarged view of a cross-section. See text for details. Note the distinction between a voxel (cluster) and a fragment.

Intensity beams are necessary to transform a cluster above the read threshold. Such high intensity, however, preclude potential aspects such as multiplexing. High intensity also limits the precision of writing because the voxel size is not small enough. Also described below are speculations of other effects high intensity may have on the memory device's writing performance efficiency:

- The photocyclic properties of the memory storage device must be improved before it can compete with current commercially available memory storage systems. Current attempts at improving quantum efficiencies[2] are underway at several institutions in a somewhat collaborative effort.
- When a cluster of bR is paged and then written to, a significant amount of the bR in that cluster must switch states in order to distinguish its bit 1 nature from a nearby unwritten bit 0 cluster. The read process must be able to distinguish a bit 0 cluster from a bit 1 cluster based on the different absorption properties of each state (Fig. 3), as well as the state change efficiency within each cluster. Using the O-state to read, as described in [1], helps reduce read-errors because the O-absorption is further away from P and Q absorption than bR absorption (see Fig. 3).

IV. I/O Bio-computing Architecture

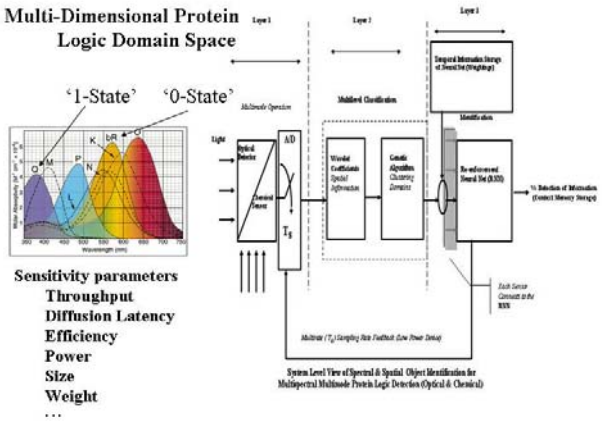


Figure 5. Overview Conversion of Multi-Dimensional Protein Logic (Layer 1,2 and 3 Stages).

If the protein bR parameters vary, but the system retains acceptable memory characteristics with disturbances, then the system is defined as robust. The general envisioned protein robust logic system consists of three stages. In stage one, the analog-to-digital (A/D) converter samples the optical protein signal and converts the sampled signal to digits within the register used in the protein computer. The polymorphic protein computer (second stage), Fig. 6, processes the A/D register by the user defined protein logic algorithms, with the result being placed in the D/A register. Stage three, with the neural network, converts the discrete D/A register data to an optical switch matrix and an optical signal.

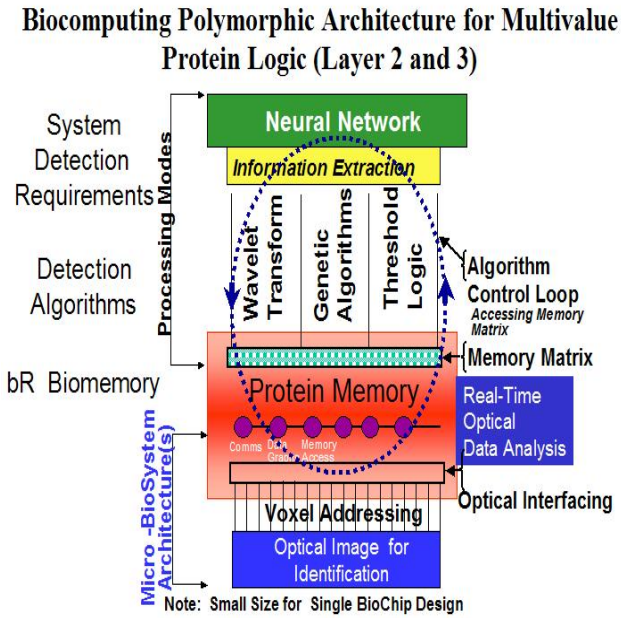


Figure 6. Details of Layer 2 and 3 –Polymorphic Biocomputing.

[2] Quantum Efficiency is defined as the ratio between the photons used for a chemical reaction and the total absorbed photons. Quantum Efficiency is sometimes interchanged with the term Quantum Yield.

Because conventional model-based methods for pattern recognition and decision-making are slow and ineffective, we have proposed a polymorphous adaptive strategy for the second stage in which combinations of the basic intelligent methodologies including *wavelet transforms, genetic algorithms, and threshold logic* are used to provide better processing capabilities to the neural network within a single system. Each of the sophisticated "intelligent" processing techniques, if used alone, possesses certain advantages and disadvantages. By including all these techniques in the polymorphic detection process, one can mix and match them in algorithms, which will be more robust and efficient over a wider range of targets than if we restricted ourselves to one technique alone. The concept of polymorphic computing, as shown in Fig. 6, refers to hardware modules with superimposed built-in functionality. A processing functional change (neural nets, genetic algorithm, and threshold logic) does not require switches, reconfiguration and separate processing modules, as in traditional approaches. Instead, the change comes from modifications in the characteristics of circuits involved in detection of the *protein logic*, in response to controls such as light response, multi-inputs, time and frequency response, sensitivity, and threshold levels. Biocomputing demonstrates the polymorphic concept and the extraction of protein logic by evolution.

V. Summary

The bio-computing design approach with bio/analog/digital integration reflects multidisplinary prototyping algorithms and 3-D structures for mult-dimensional protein logic. Realizing new advances in biocomputing, requires design advances in software algorithms equivalent to those in reconfigurable hardware. The design advances involves the use of bR protein memory and a polymorphic technology platform, that is adaptable to new biotechnology and protein logic. Cross-platform *bio/analog/digital* architecture for new hardware applications that bridges the gap between digital and multivalue logic design is a future design standard that must be addressed if biocomputing is to evolve. The need to bridge the gap between electronic devices and proteins with reconfigurable, self-healing system design introduces the concept of polymorphic computing. Once the polymorphic architectures are developed, neural networks can be used to control the polymorphic response, whereby achieving a robust and intelligent biocomputing system.

VI. References

1. Birge et al. *Biomolecular Electronics: Protein-Based Associative Processors and Volumetric Memories*. Journal of Physical Chemistry B, 1999. 103:49, p.10746-10766.
2. Stuart, J., Marcy, D., Wise, K., Birge, R. *Volumetric Optical Memory Based on Bacteriorhodopsin*. Synthetic Metals, 2002. 127, p.3-15.
3. Stuart, J. *Protein-Based Three-Dimensional Memories*. AFRL-IF-RS-TR-2001-279 Final Technical Report, January 2002. Air Force Research Laboratory, Information Directorate, Rome Research Site, Rome, New York.
4 Tallent et al. Photochemistry in Dried Polymer Films Incorporating the Deionized Blue Membrane Form of Bacteriorhodopsin. Biophysical Journal, 1998. 75, p. 1619-1634.
5. Abdel-Aty-Zohdy, Hoda S., "Neural Network and Genetic Algorithm Hardware Implementations for Communicative and Collaborative Technologies," Proc. The Collaborative Technologies Workshop, November 10–11, 1999, Rochester, MI, pp. 40-45.

APPENDIX-II

Electronic Nose Inhibition in a Spiking Neural Network for Noise Cancellation

In collaboration with: *Dr. R.L. Ewing (WPAFB/IFTA)*

ABSTRACT

Spiking neural networks can be implemented in hardware by using adders and comparators, making them well suited for real-time information extraction applications. An olfaction detection spiking neural network that detects binary odor patterns is analyzed and implemented. Detection is prone to false positive errors when multiple odors are simultaneously active. This paper presents a new method for inhibiting spiking neural networks by modulating a detection threshold. Interference noise from active odors is measured by a single inhibitory neuron. The inhibition neuron changes the detection threshold to cancel interference noise without affecting false-negative detection error. A digital implementation of the inhibition is simulated. Comparative results prove that threshold modulation reduces false-positive detection error in high noise scenarios where fifteen odors are active simultaneously.

I. Introduction

Bioinformatics classification problems are not strictly limited to gene analysis problems and also have application to models of biological sensor systems. This paper presents a system for classifying odorant patterns from an idealized electronic nose using biologically inspired spiking neural networks with a novel form of inhibition. After nearly a decade of research, pattern classification techniques for the electronic nose have created practical results [3,5,6,7,8]. Most of these solutions use computationally intense classification techniques such as principle component analysis, fuzzy logic, and neural networks. Here, we consider a simple spiking neural network in a scenario where several odors are present simultaneously, in addition to cross-contamination noise.

An objective of the electronic nose is a portable system with a very large odor memory that can be used in outdoor environments for real-time odor detection, learning, and tracking. The system we have designed has two major design constraints. First, our goal is a system-on-a-chip for real-time odor classification in a system with 1024 inputs, so we have avoided multiplication operations and created a classification system that uses only adders, comparators, and a single, small look-up table. Second, we seek to design a detection/classification system that can be expanded on a parallel data bus for real-time, simultaneous monitoring of several thousand odors.

II. A Model of Electronic Olfaction

The ideal electronic olfactory system may be modelled as an array of odorant sensors. In this model, there are $N = 1023$ unique odorant sensors. Each odorant sensor responds to a unique chemical composition. Each odorant sensor produces a binary output, and is considered 'off' until a threshold concentration level of chemical vapor turns it 'on'. Unique odors stimulate the odorant sensor array with a unique binary pattern. This model of olfaction resembles the biological model in rat brains, where odor receptor cells correspond to odorant sensors [2].

Our simulator models an odor as a random combination of n total odorant sensors in the sensor array. Binary inputs from the odor array are sampled with a rate defined as $S = 15\ Hz$, compatible with mammalian respiratory rates.

The binary outputs from the odor sensor array must be converted to spike trains that are compatible with inputs to the spiking neural network. Spike trains have an average spike-per-second rate defined by v. When the odorant sensor is in the 'off' state, spikes are input to the spiking neural network with a rate of v_{off}. When the odorant sensor is in the 'on' state, the input has a spike rate of v_{on}. This idealized representation of the electronic noise may be fairly criticized for oversimplifying the outputs of chemical sensors. However, the model is sufficiently complicated to allow us to study the effects of odor cross-contamination from an array of over 1000 sensors.

III. Odor Detection Theory

A. The Spiking Neuron for Average Spike Rate

The detection cell is a spiking neuron, which, in this system, is a very simplified approximation to an integrate and fire Neuron[4].

Spikes from the odorant sensor array serve as inputs to the neuron. When a spike is received by one of the neuron's synapses, an internal neuron counter, *V*, is incremented. When *V* reaches a neuron threshold defined by θ_v, the neuron emits a spike and *V* is reset to 0.

$$V = \begin{cases} V+1 & \text{when } V < \theta_v \text{ (and spike receipt);} \\ 0 & \text{when } V = \theta_v \text{ (Emit Spike).} \end{cases} \quad (1)$$

Now, suppose that the neuron has *c* synapses. If the i^{th} synaptic input has a spike rate of v_i, then the average output spike rate of the neuron, v_{output} is

$$\nu_{output} = \frac{\sum_{i=1}^{c} \nu_i}{\theta_v} \quad (2)$$

Equation 2 shows that v_{output} is the average of v_i in the special case when the neuron threshold equals the number of synaptic inputs, $\theta_v = c$.

B. The Spiking Neuron for Hamming Distance

Define ε as the percentage of input synapses that are 'on'. ε is the fraction of pre-synaptic inputs that are spiking with rate v_{on}. The following equations apply when $\theta_v = c$:

$$\sum_{i=1}^{c} \nu_i = \nu_{off} \cdot c \cdot (1-\epsilon) + \nu_{on} \cdot c \cdot \epsilon \quad (3)$$

$$\nu_{output} = \nu_{off} + \epsilon \cdot (\nu_{on} - \nu_{off}) \quad (4)$$

Therefore, the output spike rate of the neuron, v_{output}, is directly proportional to the fraction of pre-synaptic inputs that are 'on', ε. Hence, a neuron may be used to detect an odor by simply connecting the neuron's synaptic inputs to the unique odorant sensor combination stimulated by the odor.

C. Detection Threshold

To build tolerance in the system with a noisy environment and multiple odors, a minimum threshold value for detection, ε_{detect}, is required. For example, if $\varepsilon_{detect} = .9$, then an odor will be detected when 90% of its odorant sensors are active. The corresponding minimum output spike rate for detection is defined as v_{detect}.

$$\nu_{detect} = \nu_{off} + \epsilon_{detect} \cdot (\nu_{on} - \nu_{off}) \quad (5)$$

Thus far, the theory has focused on average spike rates, *v*. For our purposes it is more expedient to consider the period between subsequent spikes, defined as τ.

$$\tau = \frac{1}{\nu} \tag{6}$$

$$\tau_{detect} = \frac{1}{\nu_{off} + \epsilon_{detect} \cdot (\nu_{on} - \nu_{off})} \tag{7}$$

Likewise, ε is a function of τ.

$$\epsilon = \frac{1 - \tau}{\tau \cdot (\nu_{on} - \nu_{off})} \tag{8}$$

Therefore, τ may be used to estimate ε.

IV. Noise Inhibition

A. Noise

The olfaction system is seldom presented with a single odor per sniff. For example, at country fair, a single sniff of the open air simultaneously presents an assortment of odors ranging from the delightful smell of roasting corn to the smells of animals. Somehow, the olfaction system discriminates individual odors with precision. Because odorant sensors are not unique per individual odor, any number of odors might activate a particular odorant sensor. Our simulation models this effect by presenting *p* odors simultaneously during each simulated sniff.

Additionally, the simulator models salt and pepper noise, which is defined by η_{bg}, a percentage of odorant sensors randomly stimulated, and η_{bl}, a percentage of odorant sensors randomly blanked.

B. Noise Cancellation by Inhibition

To understand the effect of noise on equation 8, we decompose ε into probability based components ε_{noise} and ε_{odor}. Here, ε_{odor} is defined as the proportion of inputs that are not stimulated by noise, and ε_{noise} is the proportion of inputs that are stimulated by noise.

$$\epsilon = \epsilon_{odor} + \epsilon_{noise} - (\epsilon_{odor} \cdot \epsilon_{noise}) \tag{9}$$

Next, equation 5 is updated by equation 9.

$$\nu_{detect} = \nu_{off} + (\epsilon_{detect} + \epsilon_{noise} - (\epsilon_{detect} \cdot \epsilon_{noise})) \cdot (\nu_{on} - \nu_{off}) \tag{10}$$

Note that ε_{noise} is a system-wide phenomena, while the component of ε_{odor} is local to the synapses connected to the neuron. Because ε_{noise} is system wide, it may be estimated directly from equation 8 using a single inhibitory neuron with $c=N$ synapses connected to every odorant sensor in the system.

From equation 4, the output spike rate of this inhibitory neuron, ν_{inh}, is given when $c=N$ is large.

$$\nu_{inh} = \nu_{off} + \epsilon_{noise} \cdot (\nu_{on} - \nu_{off}) \tag{11}$$

$$\epsilon_{noise} = \frac{(\nu_{inh} - \nu_{off})}{(\nu_{on} - \nu_{off})} \tag{12}$$

Our goal is to cancel the effects of $\varepsilon_{\text{noise}}$ in the detection neurons, which is generally accomplished in biological systems by inhibitory feedback. Typically, inhibitory neurons emit spikes that decrease *V* in the detection neuron. A similar effect can be obtained by increasing θ_{v}. The approach we take here is to modulate τ_{detect}. This technique is advantageous, because negative numbers for *V* will not be possible.

First, v_{detect} is solved as a function of v_{inh} using equations 10 and 12.

$$\nu_{detect} = \epsilon_{detect} \cdot \nu_{on} + \nu_{inh} - \epsilon_{detect} \cdot \nu_{inh} \qquad (13)$$

Next, the spike rate v_{inh} is converted to a time period measurement between spikes, τ_{inh}.

$$\tau_{inh} = \frac{1}{\nu_{inh}} \qquad (14)$$

Finally, recognizing that τ_{detect} is the inverse of v_{detect}, equation 14 becomes

$$\tau_{detect} = \frac{\tau_{inh}}{\epsilon_{detect} \cdot \nu_{on} \cdot \tau_{inh} + 1 - \epsilon_{detect}} \qquad (15)$$

Therefore, equation 15 shows that the inhibitory neuron can be used to modulate τ_{detect} to cancel the system wide effects of $\varepsilon_{\text{noise}}$, thereby allowing the neuron to respond to the local effect of $\varepsilon_{\text{odor}}$.

V. The Chip Design

The network described above was implemented in Verilog for an FPGA (Field Programmable Gate Array) and then synthesized into an ASIC (Application Specific Integrated Circuit) design. Details of the circuit are outside the scope of this paper [1], although some details regarding detection and inhibition are discussed briefly.

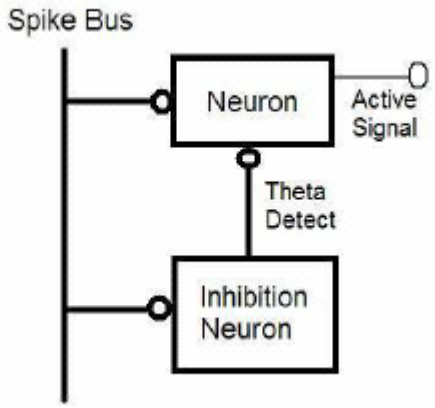


Figure 1. Circuit Block Diagram (Spikes are posted on the spike bus by odorant sensors as *10* bit addresses. The inhibition neuron monitors spikes and calculates τ_{detect} based on $\varepsilon_{\text{noise}}$. Neurons listen to spikes on the spike bus in parallel and measure the period between spikes, τ).

A. Spikes

Spikes are communicated between the odorant sensor inputs and the detection neurons by posting the address of the input on a synchronous spike-bus operating at frequency f_{sb}. The selection of spike rate effects detection speed, detection precision, and congestion on the spike bus. In our design, we selected $v_{off} = 0$ and $v_{on} = 1$, and $f_{sb} = 2$ kHz

B. Detection Neurons

The detection neuron includes a counter, V_d, and comparator to signal when $V_d = \theta_v = c$. V_d increases each time a spike is received. As shown in equation 1, V_d is reset to 0 when V_d reaches c, whereupon a decision is made about whether the odor is active or inactive.

The detection decision (deciding if the odor is active or not) uses τ_{detect}, which is compared to τ, the time period for V_d to increment from *0* to *c*. A digital implementation of this detection scheme requires another counter, χ_τ that increments with frequency f_n, to measure τ. The neuron will always be considered off (or inactive) once $> \tau_{detect}$, so χ_τ will never need to increment beyond a threshold count, θ_{detect}. We define θ_{detect} as τ_{detect} converted to the counter units of χ_τ. As discussed in the prior section, τ_{detect} (and hence θ_{detect}) will be dynamically adjusted by the inhibition neuron.

The bit width of χ_τ will depend on f_n. We chose $f_n = 200$ for a relatively precise measurement of τ while limiting χ_τ to *8* bits.

C. Inhibition

The inhibition neuron modulates τ_{detect} for the entire network of neurons, in response to noise. V_{inh} of the inhibit neuron increments whenever a spike is received by any of the *1023* inputs, and $\theta_v = c = N = 1023$. In this configuration, τ_{inh} is directly proportional to ε_{noise}, the equation below shows how τ_{detect} may be modulated as a function of τ_{inh}..

$$\theta_{detect} = \frac{\tau_{detect}}{f_n} = \frac{\tau_{inh}}{(\epsilon_{detect} \nu_{on} \tau_{inh} + 1 - \epsilon_{detect}) \cdot f_n} \quad (16)$$

To measure τ_{inh}, we define a counter χ_{inh} that will increment at frequency f_i to measure the time between successive resets of V_{inh}. χ_{inh} needs to be large because χ_{inh} tends to infinity as ε_{noise} tends to *0*. Fortunately, in a low noise environment inhibition is not necessary, so we can limit χ_{inh} to a reasonable bit count. In our final design, we choose $f_{inh} = f_{sb}/512 = 3.9$ *Hz*. This has the effect of holding χ_{inh} to *8* bits. The simulation below also compares a more precise implementation where χ_{inh} is *12* bits.

VI. Simulation and Experiment

A. Noise Scenarios

The spiking neural network was tested using three major environment scenarios to represent a broad spectrum of noise activity as shown in Table I.

1) Low Noise: In the low noise scenario, all forms of noise are present. However, the noise amount is insignificant so that there should be little if any effect on network performance.
2) Medium Noise: The medium noise scenario includes enough noise to tax the capabilities of the network, but we still expect good performance.
3) High Noise: The high noise scenario is at the edge of the network's capability. In this scenario, odor stimulus is comparable to noise stimulus. Perfect network response is not expected, but a graceful and robust degradation of network response should be apparent.

TABLE I

NOISE SCENARIO PARAMETERS: n = NUMBER OF ODORANT SENSORS STIMULATED PER ODOR. p = NUMBER OF ODORS ACTIVE SIMULTANEOUSLY DURING SIMULATION. η_{bg} = FRACTION OF INPUTS STIMULATED BY BACKGROUND NOISE. η_{bl} = FRACTION OF ODORANT SENSORS SUPPRESSED BY BLANKING NOISE.

Scenario	n	p	η_{bg}	η_{bl}
Low Noise	100	2	.05	.05
Medium Noise	100	10	.1	.1
High Noise	100	15	.15	.1

TABLE II

CHIP CONFIGURATIONS: χ_{inh} BITS = BIT WIDTH OF INHIBITION NEURON INTER-SPIKE-INTERVAL COUNTER. ϵ_{detect} = FRACTION OF ACTIVE INPUTS TO TRIGGER DETECTION WHEN NOISE IS NOT PRESENT.

Chip Configuration	**D08**	**D09**	**D18**	**D19**	**D28**	**D29**
Inhibition Enabled	no	no	yes	yes	yes	yes
χ_{inh} **bits**	NA	NA	8	8	12	12
ϵ_{detect}	.8	.9	.8	.9	.8	.9

B. Chip Configurations

Algorithms were tested using various chip design configurations summarized in Table II. Configurations ending in the number *8* have $\varepsilon_{\text{detect}} = .8$. Configurations ending in the number 9 have $\varepsilon_{\text{detect}} = .9$. For configurations D08 and D09, inhibition was disabled. Configurations D18 and D19 used *8* bit counters for χ_{inh} and $f_{inh} = 200$. Configurations D28 and D29 used *12* bit counters for χ_{inh} and $f_{inh} = 2000$.

C. Testing Procedure

A simulator written in C# allowed us to test digital chip configurations with timing identical to Verilog simulation without modifying HDL code. In a typical detection test, *300* odors were randomly generated with n = *100*. Next, *300* neurons were generated to detect these *300* odors. For example, synaptic connections of neuron *1* were loaded to detect odorant sensors stimulated by odor *1*.

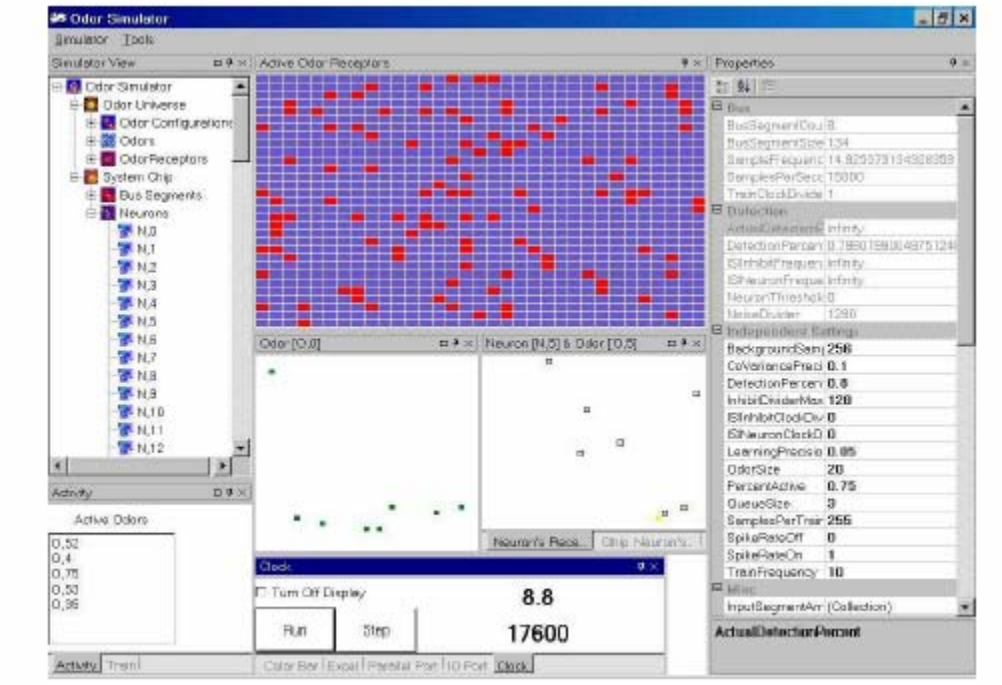


Figure 2. The odor simulator randomly generates binary odor patterns and trains simulated neurons to detect the patterns. Odors and system noise are then generated to test detection algorithms. Simulator logic is identical to FPGA digital logic circuitry.

During the test, odors were randomly cycled in and out of the test environment ensuring that exactly *p* odors were present. Odors were activated for a random period of time having a normal distribution with a mean of *3* seconds. As odors became active, neurons assigned to detect them responded, and logic in the simulator recorded detection metrics for these neurons. At the end of each test, the metrics of each neuron were recorded in a log file. Every configuration was tested in every noise scenario for *500* seconds per test. Each test was repeated *5* times.

VII. Results and Discussion

A. Results

Figures 3 to 6 summarize the detection error sorted by noise scenario (H=high noise, L=low noise, and M= medium noise). Chip configuration codes are listed in Table II.

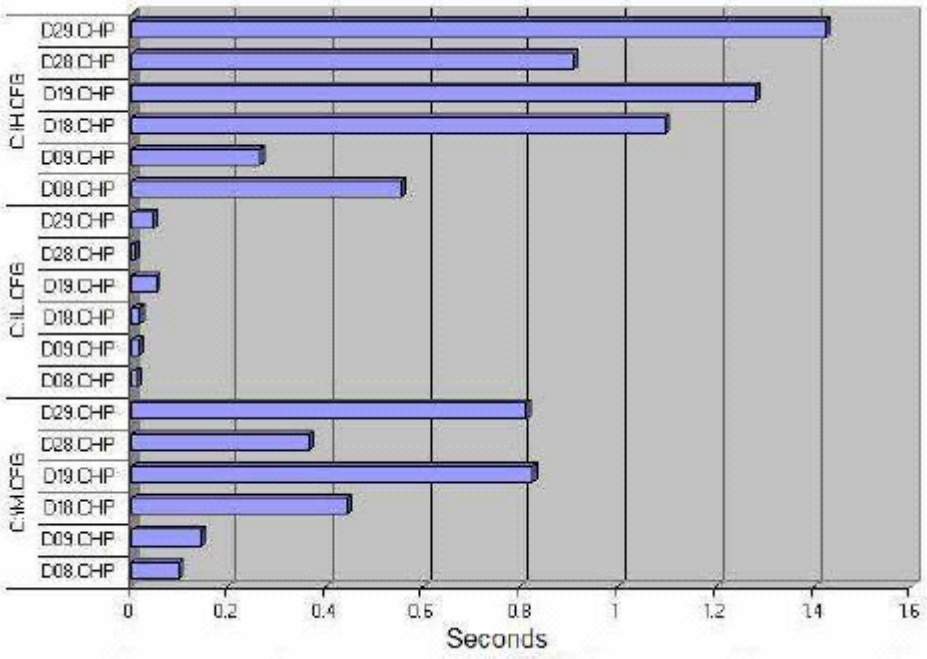


Figure 3. Detection Error: Detection error is a measure of false-negative response from neurons, and measures the total time when a neuron does not detect an active odor. The detection error is measured by average number of seconds of error per detection neuron. Because the total number of seconds each odor was present differs for each noise scenario, direct comparisons between H, L, and M are not helpful. However, comparisons within each noise scenario show that inhibition increases detection error.

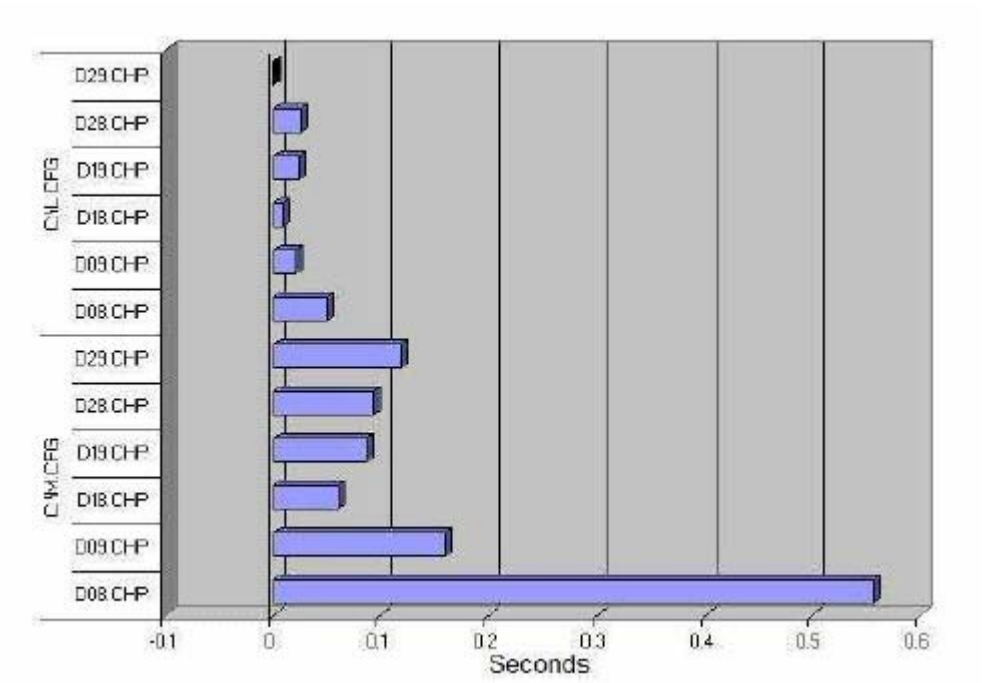


Figure 4. Silence Error in Low and Medium Noise: The silence error is the average number of seconds when a detection neuron falsely detects an odor that is not active. Silence error is a measure of false-positive detections. Again, direct comparisons between H, L, and M are not helpful because odors were present for different lengths of time in each scenario. Inhibition reduces the effect of false-positive detections, although the effect is least pronounced in a low noise scenario.

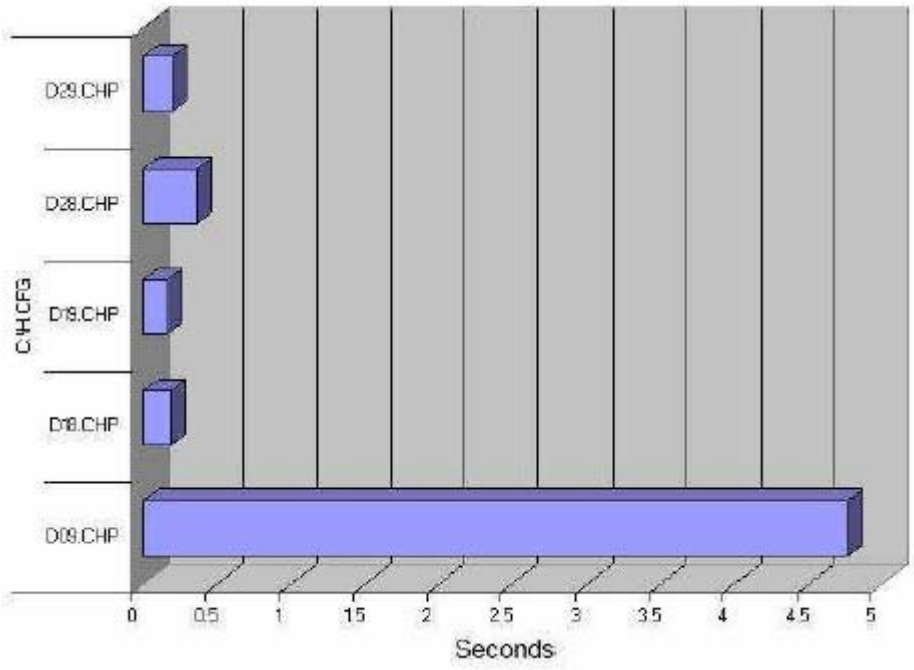


Figure 5. Silence Error in High Noise: False-positive detections are significantly reduced by inhibition in the high noise scenario. Note: The D08 chip configuration without inhibition has an error exceeding 200 seconds and was omitted because it dwarfed the other entries.

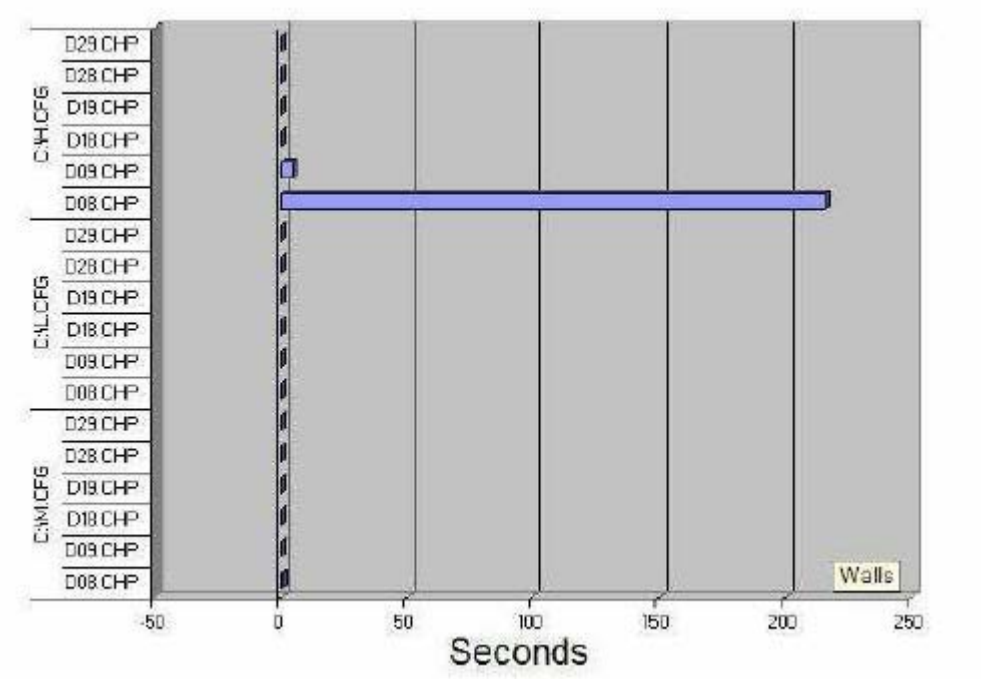


Figure 6. Silence Error Summary: This view emphasizes the D08 chip configuration's false-positive error in the high noise scenario.

B. Discussion

In the low noise scenario, there is no substantial difference between configurations that implement inhibition and those without inhibition. However, in high noise scenarios, false-positive error increases dramatically, rendering the system non-functional for the D08 configuration (without inhibition). Conversely, configurations implementing

inhibition cancelled the effect of system-wide noise to allow individual odors to be identified despite interference from other odors simultaneously active in the system.

The detection error illustrates the downside to this inhibition scheme, because configurations implementing inhibition are more prone to false-negative errors. However, the trade-off is acceptable because false-positive error is least desirable for odor detection. The results show a system that degrades gracefully in the presence of noise by simply taking longer to detect odors. Indeed, the inhibition causes the detection to simply become more conservative in a high noise scenario, while the network without inhibition becomes unusable.

Finally, there is noticeable, but not critical, difference between results from the *8* bit and the *12* bit counter for χ_{inh}. Also, differences in result when $\varepsilon_{\text{detect}}$ is *.8* or *.9* are most significant when inhibition is not present in a high noise scenario.

VIII. Conclusion

In conclusion, we have presented a new technique for modulating inhibition for a spiking neural network. This method allows the network to detect individual odors when several odors are simultaneously present. The results show that configurations without inhibition produce abnormally high false-positive error, while the performance of chips with inhibition remain robust and degrade gracefully in the presence of noise.

Furthermore, this inhibition technique is well suited for digital logic implementation because a single inhibition neuron modulates the detection threshold, τ_{detect}, for the entire network. All other detection neurons may be implemented in simple circuits based on counters and comparators.

Finally, this paper focuses on an electronic nose application, but this approach to binary pattern sorting has potential use in other bioinformatic applications that require parallel sorting of competing/overlapping binary patterns.

IX. References

[1] J. Allen, H. Abdel-Aty-Zohdy, R. Ewing, T. Chang Spiking Networks for Biochemical Detection. *Proc. 45th IEEE Int'l Midwest Symposium on Circuits and Systems (MWSCS 02)*, Aug. 2002.

[2] L. Buck The Molecular Architecture of Odor and Pheromone Sensing in Mammals. *Cell*, 100:611-618, March 5, 1996.

[3] R. Dutta, K.R. Kashwan, M. Bhuyan, E.L. Hines, J.W. Gardner Electronic nose based tea quality standardization. *Neural Networks*, 16:847-853, 2003.

[4] S. Fusi *Neural Networks: function and learning in theory and experiment*, doctoral dissertation, Dept. of Physics, Jerusalem University, Jerusalem, Isreal. Aug. 1999.

[5] B. G. Kermani, S. Schiffman, H. Nagle Using Neural Networks and Genetic Algorithms to Enhance Performance in an Electronic Nose. *IEEE Trans. on Biomedical Engineering*, 46:429-438, Apr. 1999.

[6] R. Youssif, C. Purdy A Multistrategy Signal Pattern Classifier. *Proc. 45th IEEE Int'l Midwest Symposium on Circuits and Systems (MWSCAS 02)*, Aug. 2002.

[7] O. Rochel, D. Martinex, E. Hughes, and F. Sarry Stero-olfaction with a Sniffing Neuromorphic Robot Using Spiking Neurons. *Eurosensors*, 2002.

[8] H. S. Abdel-Aty-Zohdy Reinforcement Neural Learning With Application To Gas Sensors. *Proc. 40th IEEE Int'l Midwest Symposium on Circuits and Systems (MWSCS 98)*, Aug. 1997.